



## **TUGAS AKHIR - TE 145561**

### **RANCANG BANGUN PENGENDALI UTAMA PADA SISTEM KONTROL DAN MONITORING LAMPU PENERANGAN JALAN UMUM TENAGA SURYA MENGGUNAKAN *WIRELESS SENSOR NETWORK* DENGAN HUMAN MACHINE INTERFACE TERPUSAT**

Raka Satria Pradana H.D.  
NRP 2214 030 094

Dosen Pembimbing  
Suwito, ST.,MT.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 145561**

***DESIGN AND BUILD FIRST CONTROLLER IN CONTROL SYSTEM AND  
MONITORING PUBLIC STREET LIGHTING OF SOLAR ENERGY USING  
WIRELESS SENSOR NETWORK WITH HUMAN MACHINE INTERFACE***

Raka Satria Pradana H.D.  
NRP 2214 030 094

Advisor  
Suwito, ST.,MT.

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**RANCANG BANGUN PENGENDALI UTAMA PADA SISTEM KONTROL DAN MONITORING LAMPU PENERANGAN JALAN UMUM TENAGA SURYA MENGGUNAKAN *WIRELESS SENSOR NETWORK* DENGAN HUMAN MACHINE INTERFACE TERPUSAT**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, **17 July 2017**

Raka Satria Pradana H.D  
NRP 2214030094

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN PENGENDALI UTAMA PADA SISTEM  
KONTROL DAN MONITORING LAMPU PENERANGAN  
JALAN UMUM TENAGA SURYA MENGGUNAKAN *WIRELESS*  
*SENSOR NETWORK* DENGAN *HUMAN MACHINE INTERFACE*  
TERPUSAT**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Program Studi Komputer Kontrol  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:  
Dosen Pembimbing

Suwito, ST., MT.

NIP. 1959 02 18 1986 10 1001

**SURABAYA,  
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----



# **RANCANG BANGUN PENGENDALI UTAMA PADA SISTEM KONTROL DAN MONITORING LAMPU PENERANGAN JALAN UMUM TENAGA SURYA MENGGUNAKAN *WIRELESS SENSOR NETWORK* DENGAN HUMAN MACHINE INTERFACE TERPUSAT**

**Nama : Raka Satria Pradana H.D.**  
**Pembimbing : Suwito, ST.,MT.**

## **ABSTRAK**

Penerangan Jalan Umum Tenaga Surya (PJU-TS) adalah penerangan jalan umum dimana daya listrik disuplai secara mandiri yang diperoleh dari energi matahari. Saat ini belum ada sistem yang digunakan untuk mengendalikan dan memonitoring kondisi PJU-TS. Kondisi seperti ini efektif menggunakan Komunikasi *wireless* yang diintegrasikan dengan *Wireless Sensor Network*. Sehingga, Lampu PJU-TS dapat diMonitoring secara terpusat. Dalam pembuatan Tugas Akhir ini digunakan sistem *embedded* Mikrokontroler dengan tatap muka *Human Machine Interface*. Media Komunikasi yang digunakan adalah *Transceiver* NRF24L01+. Tugas Akhir ini membahas masalah Pengendali Utama pada sistem Kontrol dan Monitoring lampu PJU-TS. Pengendali utama ini bertugas sebagai Master untuk Kontrol masing-masing Lampu PJU-TS dan Monitoring data yang dikirimkan oleh Slave, Data tersebut berupa nilai Tegangan dari Solar Cell dan Arus dari Lampu PJU-TS. Data tersebut kemudian disimpan di Datalogger. Dari hasil pengujian didapatkan bahwa NRF24L01+ dapat terintegrasi dengan *Wireless Sensor Network* sehingga data dapat dikirimkan sejauh 60 meter dengan presentase keberhasilan Kontrol sebesar 100 % dan Monitoring dengan kondisi lingkungan terbuka sebesar 100% dalam jangkauan 20 meter. Dari hasil tersebut, *Wireless Sensor Network* dapat mengawasi dan mengendalikan kerja Lampu PJU-TS secara terpusat.

**Kata Kunci :** Penerangan Jalan Umum Tenaga Surya (PJU-TS), *Wireless Sensor Network* (WSN), *Human Machine Interface*, Datalogger, Mikrokontroler.

-----Halaman ini sengaja dikosongkan-----

**DESIGN AND BUILD FIRST CONTROLLER IN CONTROL  
SYSTEM AND MONITORING PUBLIC STREET LIGHTING OF  
SOLAR ENERGY USING WIRELESS SENSOR NETWORK  
WITH HUMAN MACHINE INTERFACE**

**Name : Raka Satria Pradana H.D.**

**Advisor : Suwito, ST.,MT.**

**ABSTRACT**

*Wireless Sensor Network (WSN) is an integrated wireless network consisting of a set of sensor nodes scattered in an area, aiming for control and Monitoring of the area. Using WSN can solve the problem of remote Monitoring without using a wide range of wireless communications media with great power. The communication medium used is Transceiver NRF24L01 +. This Final Project discusses the main controller problem in control system and Monitoring of PJU-TS lamp. The main controller is assigned as Master for Control of each PJU-TS Light and Monitoring data transmitted by Slave, The data is Voltage value of Solar Cell and Flow of PJU-TS Lamp sent by Slave in each PJU-TS. The data is then stored in Datalogger. While the Control and Monitoring commands can be done through the Human Machine Interface. From the test results found that NRF24L01 + can be integrated with Wireless Sensor Network so that data can be sent as far as 60 meters with 95,33% control success percentage and Monitoring condition of PJU-TS open area lamp by 50% LOSS environmental conditions within 20 meters. From that result, Wireless Sensor Network can monitor and control the work of PJU-TS Lamp centrally.*

**Keyword :** *Wireless Sensor Network (WSN), Human Machine, Interface, Datalogger, Public Street Lighting of Solar energy*

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

Alhamdulillah kami panjatkan kepada Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan kemudahan dariNya, hingga kami dapat menyelesaikan Tugas Akhir ini dengan baik, begitu pula dengan pembuatan buku Tugas Akhir ini.

Tugas akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

**Rancang Bangun Pengendali Utama pada Sistem Kontrol dan Monitoring Lampu Penerangan Jalan Umum Tenaga Surya menggunakan *Wireless Sensor Network* yang dengan *Human Machine Interface* Terpusat.**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Suwito, ST., MT. Atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 8 Juni 2015

Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR .....	v
HALAMAN PENGESAHAN .....	vii
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan .....	3
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Laporan.....	4
1.7 Relevansi .....	4
<b>BAB II TEORI DASAR .....</b>	<b>7</b>
2.1 Mikrokontroler ATmega 2560 .....	7
2.1.1 Memori.....	8
2.1.2 Input dan Output.....	8
2.1.3 Spesifikasi Mikrokontroler ATmega 2560.....	9
2.2 Mikrokontroler ATmega 328.....	9
2.3 Bluetooth.....	10
2.4 Media Penyimpanan.....	11
2.5 <i>Real Time Clock</i> .....	12
2.6 Serial RS-232 .....	13
2.7 Komunikasi <i>Wireless</i> .....	16
2.8 <i>Wireless Sensor Network (WSN)</i> .....	17
2.8.1 Pengertian dan Perkembangan WSN.....	17
2.8.2 Arsitektur WSN.....	18

2.8.3	Bagian-Bagian WSN.....	19
2.9	HMI ( <i>Human Machine Interface</i> ).....	20
2.9.1	WienView Weintek MT8070iH.....	20
2.10	Protokol Modbus.....	22
BAB III PERANCANGAN SISTEM KONTROL.....		25
3.1	Perancangan Perangkat Keras.....	27
3.1.1	Rangkaian Mikrokontroler ATmega 2560.....	29
3.1.2	Rangkaian <i>Power Supply</i> .....	31
3.1.3	Rangkaian NRF24L01+ .....	32
3.1.4	Rangkaian Serial RS-232 .....	34
3.1.5	Rangkaian <i>Real Time Clock</i> Ds1307.....	35
3.1.6	Rangkaian Bluetooth HC-05 .....	36
3.1.7	Rangkaian Arduino Nano.....	37
3.1.8	Rangkaian Catalex SD Card .....	39
3.1.9	HMI ( <i>Human Machine Interface</i> ).....	41
3.2	Perancangan Perangkat Lunak.....	42
3.2.1	Metode Pengiriman Menggunakan <i>Wireless Sensor Network</i> .....	43
3.2.2	Alogaritma pada Arduino Mega 2560.....	45
3.2.3	Alogaritma pada Arduino Nano .....	51
3.2.4	Setup Weintek MT8070iH.....	54
BAB IV HASIL SIMULASI DAN IMPLEMENTASI .....		59
4.1	Pengujian Jarak NRF24L01+.....	59
4.2	Pengujian Kontrol Lampu PJU-TS .....	61
4.3	Pengujian Monitoring Lampu PJU-TS .....	63
4.4	Pengujian Mode Otomatis “Kontrol” Lampu PJU-TS.....	65
4.5	Pengujian <i>Human Machine Interface</i> MT8070iH.....	66
4.6	Pengujian Datalogger Lampu PJU-TS.....	67
BAB V PENUTUP .....		69
DAFTAR PUSTAKA .....		71
LAMPIRAN A .....		73
LAMPIRAN B.....		103
LAMPIRAN C.....		127
RIWAYAT HIDUP PENULIS .....		141



## DAFTAR GAMBAR

### HALAMAN

<b>Gambar 2.1</b>	Pemetaan Pin ATmega 2560 .....	7
<b>Gambar 2.2</b>	Pemetaan Pin ATmega 328 .....	9
<b>Gambar 2.3</b>	Bluetooth HC-05 .....	11
<b>Gambar 2.4</b>	Catalex SD Card .....	12
<b>Gambar 2.5</b>	RTC Ds1307 .....	13
<b>Gambar 2.6</b>	Level Tegangan RS-232 .....	14
<b>Gambar 2.7</b>	Konektor DB9 .....	15
<b>Gambar 2.8</b>	Serial RS-232 .....	16
<b>Gambar 2.9</b>	Modul NRF24L01+ .....	17
<b>Gambar 2.10</b>	Arsitektur WSN .....	19
<b>Gambar 2.11</b>	Dimensi MT8070iH .....	20
<b>Gambar 3.1</b>	Slave Sistem Lampu PJU-TS .....	25
<b>Gambar 3.2</b>	Master Sistem Lampu PJU-TS .....	26
<b>Gambar 3.3</b>	Ilustrasi Proses Pengiriman Data .....	26
<b>Gambar 3.4</b>	Diagram Bagian Master .....	28
<b>Gambar 3.5</b>	Rangkaian Arduino Mega 2560 .....	30
<b>Gambar 3.6</b>	Fungsi Pin Arduino Mega .....	30
<b>Gambar 3.7</b>	Rangkaian <i>Power Supply</i> .....	32
<b>Gambar 3.8</b>	Rangkaian <i>Transceiver</i> NRF24L01+ .....	33
<b>Gambar 3.9</b>	<i>Wiring</i> Pin <i>Transceiver</i> NRF24L01+ .....	34
<b>Gambar 3.10</b>	Rangkaian Serial MAX-232 .....	35
<b>Gambar 3.11</b>	Rangkaian RTC Ds1307 .....	36
<b>Gambar 3.12</b>	<i>Wiring</i> Bluetooth HC-05 .....	37
<b>Gambar 3.13</b>	Rangkaian Arduino Nano .....	38
<b>Gambar 3.14</b>	Fungsi pin Arduino Nano .....	39
<b>Gambar 3.15</b>	Rangkaian Catalex SD Card .....	40
<b>Gambar 3.16</b>	<i>Wiring</i> Catalex SD Card .....	41
<b>Gambar 3.17</b>	HMI WienView MT8070iH .....	42
<b>Gambar 3.18</b>	<i>Wiring</i> HMI MT8070iH .....	42
<b>Gambar 3.19</b>	Metode <i>Wireless Sensor Network</i> .....	43
<b>Gambar 3.20</b>	<i>Sketch</i> Paket Data Sensor .....	44
<b>Gambar 3.21</b>	Paket Data .....	44
<b>Gambar 3.22</b>	Pipe <i>Address</i> NRF24L01 .....	45

<b>Gambar 3.23</b>	<i>Sketch Transmitter NRF24L01+</i> .....	45
<b>Gambar 3.24</b>	<i>Sketch Receiver NRF24L01+</i> .....	45
<b>Gambar 3.25</b>	<i>Flowchart Main Program Arduino Mega</i> .....	46
<b>Gambar 3.26</b>	<i>Sketch Main Program Arduino</i> .....	47
<b>Gambar 3.27</b>	<i>Flowchart Fungsi Manual Arduino Mega</i> .....	48
<b>Gambar 3.28</b>	<i>Sketch Kontrol Fungsi Manual Arduino Mega</i> .....	49
<b>Gambar 3.29</b>	<i>Sketch Monitoring Arduino Mega</i> .....	49
<b>Gambar 3.30</b>	<i>Flowchart Fungsi Otomatis Arduino Mega</i> .....	50
<b>Gambar 3.31</b>	<i>Software Easy Builder 8000</i> .....	55
<b>Gambar 3.32</b>	<i>Project Manager EB8000</i> .....	56
<b>Gambar 3.33</b>	<i>Setup Easy Builder 8000</i> .....	56
<b>Gambar 3.34</b>	<i>Sistem Parameter Setting EB8000</i> .....	57
<b>Gambar 3.35</b>	<i>Device Properties EB8000</i> .....	58
<b>Gambar 3.36</b>	<i>Tampilan Screen EB8000</i> .....	58
<b>Gambar 3.37</b>	<i>Screen 1</i> .....	60
<b>Gambar 3.38</b>	<i>Screen 2</i> .....	60
<b>Gambar 3.39</b>	<i>Screen 3</i> .....	61
<b>Gambar 3.40</b>	<i>Screen 3</i> .....	61
<b>Gambar 4.1</b>	<i>Pengujian Jarak NRF24L01+</i> .....	60
<b>Gambar 4.2</b>	<i>Pengujian Kontrol Lampu PJU-TS</i> .....	62
<b>Gambar 4.3</b>	<i>Pengujian Monitoring Lampu PJU-TS</i> .....	64
<b>Gambar 4.4</b>	<i>Human Machine Interface MT8070iH</i> .....	66
<b>Gambar 4.5</b>	<i>Device tidak Terkoneksi dengan HMI</i> .....	67
<b>Gambar 4.6</b>	<i>Datalogger SD Card</i> .....	68
<b>Gambar 4.7</b>	<i>Hasil Datalogger SD Card</i> .....	68
<b>Grafik 4.1</b>	<i>Pengujian Jarak NRF24L01+ (Tanpa Halangan)</i> .....	60
<b>Grafik 4.2</b>	<i>Pengujian Jarak NRF24L01+ (Terdapat Halangan)</i> .....	61
<b>Grafik 4.2</b>	<i>Pengujian Monitoring</i> .....	63

## DAFTAR TABEL

### HALAMAN

<b>Tabel 2.1</b> Spesifikasi Mikrokontroler ATmega 2560.....	9
<b>Tabel 2.2</b> Spesifikasi Mikrokontroler ATmega 328.....	10
<b>Tabel 2.3</b> Fungsi Pin DB9 .....	15
<b>Tabel 2.4</b> Register Protokol Modbus.....	23
<b>Tabel 3.1</b> Fungsi Pin Arduino Mega 2560 .....	31
<b>Tabel 3.2</b> <i>Wiring</i> NRF24L01+ dengan Arduino Mega 2560.....	34
<b>Tabel 3.3</b> <i>Wiring</i> RS-232 dengan Arduino Mega 2560.....	35
<b>Tabel 3.4</b> <i>Wiring</i> Ds1307 dengan Arduino Mega 2560.....	36
<b>Tabel 3.5</b> <i>Wiring</i> Bluetooth HC-05 dengan Arduino Mega 2560.....	37
<b>Tabel 3.6</b> <i>Wiring</i> Arduino Nano 328 dengan Arduino Mega 2560 .....	39
<b>Tabel 3.7</b> <i>Wiring</i> Catalex SD Card dengan Arduino Nano 328.....	41
<b>Tabel 3.8</b> <i>Address</i> Holding Register Modbus.....	59
<b>Tabel 4.1</b> Kontrol dari Master ke Slave 1 .....	62
<b>Tabel 4.2</b> Kontrol dari Master ke Slave 2.....	62
<b>Tabel 4.3</b> Kontrol dari Master ke Slave 3.....	63
<b>Tabel 4.4</b> Kontrol Otomatis dari MASTER ke SLAVE 1 .....	65

-----Halaman ini sengaja dikosongkan-----

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Penerangan Jalan Umum Tenaga Surya (PJU-TS) adalah lampu penerangan jalan yang menggunakan cahaya matahari sebagai sumber energi listriknya [3]. Dalam 5 (lima) tahun terakhir, PJU tenaga surya begitu populer di kalangan pemerintah daerah sebagai alternatif untuk mengurangi biaya operasi khususnya biaya listrik PJU [8]. Pemikiran sederhana bahwa dengan sumber tenaga listrik dari panel surya, maka akan menghilangkan biaya listrik PJU dan berarti akan lebih murah.

Akan tetapi untuk menghasilkan listrik yang optimal, panel surya harus rutin dipelihara. Pembersihan panel harus rutin dilakukan sehingga kemungkinan sinar matahari terhalang oleh debu dapat diminimalkan [8]. Mengingat kondisi iklim dan kondisi jalan di Indonesia, maka pemeliharaan ini cukup merepotkan dari segi biaya (alokasi SDM dan peralatan) dan segi waktu untuk mendapatkan energi yang didapatkan maksimal [8].

Secara umum PJU tenaga surya dilengkapi dengan sensor LDR. Dengan adanya sensor LDR, lampu secara otomatis menyala pada malam hari dan mati pada pagi hari. Dengan menggunakan sensor LDR memiliki kelemahan yakni lampu akan menyala secara keseluruhan jika kondisi lingkungan mendung dan tidak ada penyeleksian (on/off) lampu pada saat sore menjelang malam. Hal ini menyebabkan konsumsi daya lampu PJU Tenaga Surya tidak hemat energi.

Untuk mengatasi permasalahan tersebut, dibutuhkan pengendalian sekaligus pemantauan peralatan Penerangan Jalan Umum Tenaga Surya (PJU-TS) dengan menggunakan Mikrokontroler. Sistem Mikrokontroler ini dapat memberikan informasi terkait kondisi (on/off) lampu dan kondisi arus yang masuk pada baterai PJU-TS dan dapat melakukan pengendalian setiap lampu PJU-TS secara terpusat. Sistem ini dibuat ke dalam 2 bagian yaitu Master dan Slave. Master adalah Bagian yang bertugas untuk melakukan Kontrol dan Monitoring sensor, sedangkan Slave bertugas untuk mengukur tegangan dari Solar Cell dan Arus Lampu PJU-TS kemudian mengirimkannya kepada Master.

Berdasarkan latar belakang tersebut, menarik bagi penulis dalam penyusunan Tugas Akhir dengan judul “Rancang Bangun Pengendali

Utama pada Sistem Kontrol dan Monitoring Lampu PJU-TS Menggunakan *Wireless Sensor Network* dengan HMI (Human Machine Interface) Terpusat”.

Pada bagian ini terdapat 2 mode pengendalian yakni Mode Manual dan Mode Otomatis/Terjadwal. Mode manual dilakukan untuk mengendalikan Lampu PJU-TS sesuai perintah Operator atau kondisi cuaca, misalnya pada saat Kabut dan Hujan. Proses pengendalian dan pemantauan lampu PJU-TS pada Tugas Akhir ini, dapat dilakukan secara terpusat melalui Human Machine Interface. Besarnya data yang dapat dipantau melalui HMI adalah Tegangan Solar Cell dan Arus Lampu PJU-TS. Komunikasi yang digunakan untuk berkomunikasi antara Master dengan Slave menggunakan Komunikasi *Wireless* dengan NRF24L01+. Metode yang digunakan untuk mengkonfigurasi antara Master dengan Slave menggunakan *Wireless Sensor Network*, Jadi Masing-masing Slave terintegrasi dengan Master serta dapat memberikan informasi kondisi Lampu PJU-TS. Selain itu data hasil pembacaan Oleh Master dilakukan pencatatan dan perekaman data yang disimpan di dalam SD Card guna mempermudah dalam hal perawatan dan perbaikan.

## **1.2 Permasalahan**

PJU tenaga surya begitu populer sebagai alternatif untuk lampu PJU konvensional, Karena memiliki banyak keuntungan yaitu dalam hal : Ramah lingkungan dan bebas polusi, sumber energi melimpah yang tak terbatas, tidak tergantung jaringan PLN, Nihil biaya listrik PLN, dapat dipasang dimana saja dan memiliki usia pakai yang panjang. Namun dari banyaknya kelebihan yang di tawarkan oleh PJU-TS. Namun, memiliki kekurangan yaitu tergantung cuaca dan komponen utama yang perlu dilakukan perawatan dan pemeliharaan, seperti Kebutuhan Daya Baterai dan efisiensi Solar Cell. Untuk itu diperlukan suatu sistem yang dapat mengendalikan kondisi on/off PJU-TS sesuai kebutuhan dan memantau Tegangan Solar Cell dan Arus Lampu PJU-TS menggunakan *Wireless Sensors Network*.

## **1.3 Batasan Masalah**

Batasan masalah dari Sistem Pengendali Lampu Penerangan Jalan Umum (PJU) menggunakan *Wireless Sensor Network* adalah :

1. Aktifitas Kontrol dan Monitoring Dilakukan Pada Weintek WienView MT8070iH dan Android

2. Jangkauan Komunikasi *wireless* sejauh 60 Meter
3. Data Hasil Pembacaan sensor oleh Master Disimpan dalam SD Card
4. Data yang di Monitoring berupa Tegangan dari Solar Cell dan Arus pada Lampu PJU-TS
5. Data yang di kirim Oleh Slave sesuai dengan interval waktu yang telah di tentukan.

#### **1.4 Tujuan**

Tujuan utama Tugas Akhir ini adalah mengembangkan dari sistem pengendalian dan Monitoring Penerangan Jalan Umum Tenaga Surya (PJU-TS). Dimana sistem ini saling terintegrasi antara masing-masing PJU dengan menggunakan media *wireless* dengan daya rendah, dan dapat dilakukan pengendalian kepada masing-masing Lampu PJU-TS. Sehingga dapat melakukan aktifitas Kontrol dan Monitoring Lampu PJU-TS dapat dilakukan secara terpusat menggunakan Human Machine Interface oleh Operator. Sedangkan, tujuan dilakukan pengendalian dan Monitoring diharapkan dapat mengawasi dan mengendalikan Kerja Lampu PJU-TS secara terpusat.

#### **1.5 Metodologi Penelitian**

Dalam pembuatan Tugas Akhir Rancang Bangun Pengendali Utama pada Sistem Kontrol dan Monitoring Lampu PJU-TS menggunakan *Wireless Sensor Network* dengan HMI terpusat. Ada beberapa tahap kegiatan yaitu meliputi tahap persiapan (*study literature*), tahap perencanaan dan pembuatan alat, tahap pengujian dan analisa, serta penyusunan laporan.

Pada tahap *study literature* akan dipelajari mengenai konsep dasar Lampu PJU Tenaga Surya kelebihan maupun kekurangan, mempelajari Komunikasi menggunakan media *wireless*, mempelajari pemrograman dengan Arduino IDE, mempelajari Mikrokontroler ATmega 2560 dan 328, mempelajari cara parsing data dan membuat paket data pada String.

Tahap perencanaan dan pembuatan alat meliputi perencanaan pembuatan perangkat keras seperti rangkaian power supply, Kontroler, *Real Time Clock* dan Serial RS-232. Melakukan konfigurasi Komunikasi sistem secara keseluruhan.

Setelah itu dilakukan pengujian alat, menganalisa kesalahan atau kegagalan pada alat dan mengatasi permasalahan tersebut. Tahapan ini dilakukan dengan melakukan pengujian Komunikasi Antara Master

dengan Slave, Integrasi Mikrokontroler dengan Human Machine Interface dan pengujian jarak Komunikasi *wireless*. Data hasil pengujian tersebut akan dianalisa kemudian mencari tahu faktor apa saja yang menyebabkan alat tidak bekerja sesuai dengan keinginan atau terjadi error. Tahap akhir penelitian adalah penyusunan laporan penelitian.

## **1.6 Sistematika Laporan**

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### **Bab I      Pendahuluan**

Membahas tentang latar belakang, permasalahan, tujuan, batasan masalah, sistematika penulisan serta relevansi yang digunakan dalam Tugas Akhir yang dibuat.

### **Bab II     Teori Dasar**

Menjelaskan dasar teori yang berisi tentang konsep yang dijadikan landasan dan mendukung dalam perencanaan serta pembuatan alat yang dibuat.

### **Bab III    Perancangan Sistem Kontrol**

Dalam bab ini membahas tentang perencanaan dan pembuatan perangkat keras (*hardware*) yang terdiri atas rangkaian elektronika dan perangkat lunak (*software*) yang terdiri atas program yang akan digunakan untuk menjalankan alat tersebut.

### **Bab IV    Pengujian dan Analisa**

Membahas tentang pengujian alat dan analisa data yang didapat dalam pengujian alat.

### **Bab V     Penutup**

Berisi tentang kesimpulan alat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini selanjutnya.

## **1.7 Relevansi**

Manfaat dari Tugas Akhir ini adalah untuk memudahkan dalam hal pengawasan kondisi Lampu PJU-TS secara terpusat sehingga memudahkan dalam hal perawatan dan perbaikan tanpa perlu pergi ke jalan untuk mengecek dan mengukur Lampu PJU-TS. Selain itu diharapkan dapat mengoptimalkan pemakaian daya dari Lampu PJU-TS



sesuai dengan kondisi cuaca ataupun waktu dengan pengendalian Lampu PJU-TS, Sehingga penggunaan Lampu PJU-TS bisa dioptimalkan terlebih Lampu PJU-TS memiliki Supply daya mandiri sehingga tidak tergantung pada jaringan listrik PLN.

-----Halaman ini sengaja dikosongkan-----

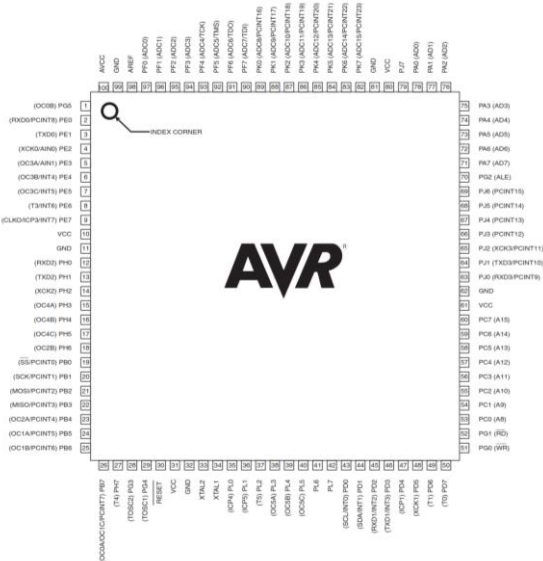
## BAB II TEORI DASAR

Untuk memahami persoalan yang dibahas pada implementasi Sistem kontrol dan Monitoring Lampu PJU-TS menggunakan *wireless sensor network* menggunakan *Human Machine Interface* (HMI) terpusat. Uraian teori terdiri dari Mikrokontroler ATmega 2560, Mikrokontroler ATmega 328, Media Penyimpanan, *Real Time Clock*, Serial RS-232, Komunikasi *Wireless*, *Wireless Sensor Network*, Protokol MODBUS.

### 2.1 Mikrokontroler ATmega 2560

Mikrokontroler ATmega 2560 adalah papan mikrokontroler berbasis ATmega 2560). ATmega 2560 memiliki 54 pin digital input/output, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input analog*, dan 4 pin sebagai UART (*port serial hardware*), 16 MHz kristal osilator. Mikrokontroler ATmega 2560 ditunjukkan pada Gambar 2.1.

Berikut ini adalah pemetaan pin Mikrokontroler ATmega 2560:



**Gambar 2.1** Pemetaan Pin ATmega 2560

### 2.1.1 Memori

ATMega 2560 memiliki 256 KB flash memori untuk menyimpan kode (yang 8 KB digunakan untuk bootloader), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

### 2.1.2 Input dan Output

Masing-masing dari 54 digital pin pada ATMega dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. ATMega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara *default*) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

1. **Serial** : 0 (RX) dan 1 (TX); **Serial 1** : 19 (RX) dan 18 (TX); **Serial 2** : 17 (RX) dan 16 (TX); **Serial 3** : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL.
2. **Eksternal Interrupt** : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2).
3. **SPI** : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung Komunikasi SPI menggunakan perpustakaan SPI.
4. **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung Komunikasi TWI menggunakan perpustakaan Wire.

ATMega 2560 memiliki 16 pin sebagai *analog input*, yang masing-masing menyediakan resolusi 10 bit. Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analog reference*.

Ada beberapa pin lainnya yang tersedia, antara lain:

1. **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
2. **RESET** : Jalur *LOW* ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada *shield*.

### 2.1.3 Spesifikasi Mikrokontroler ATmega 2560

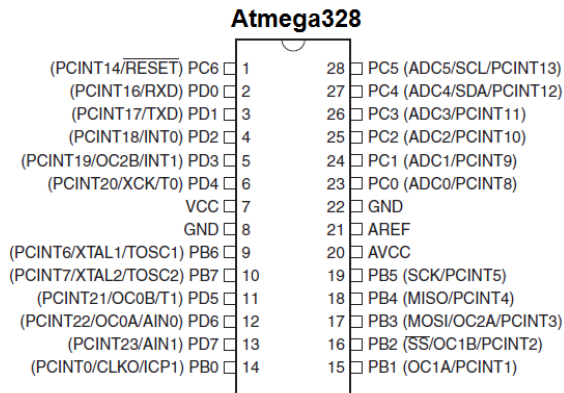
Berikut ini adalah spesifikasi dari Mikrokontroler ATmega 2560 :

**Tabel 2.1** Spesifikasi Mikrokontroler ATmega 2560

Mikrokontroler	ATmega 2650
Tegangan Operasi	5V
Input Voltage	7-12V
Input Volatge (limit)	6-2-V
Pin Digital I/O	54 (15 pin digunakan sebagai output PWM)
Pin Input Analog	16
Arus DC per pin I/O	40Ma
Arus DC untuk pin 3.3V	50mA
Flash Memori	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

### 2.2 Mikrokontroler ATmega 328

Mikrokontroler ATmega 328 memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal. Mikrokontroler ATmega 328 ditunjukan pada Gambar 2.2. Sedangkan, Spesifikasi Mikrokontroler ATmega 328 dapat dilihat pada Tabel 2.2.



**Gambar 2.2** Pemetaan Pin ATmega 328

Deskripsi Mikrokontroler ATmega 328 :

**Tabel 2.2** Spesifikasi Mikrokontroler ATmega 328

Mikrokontroler	ATmega 328
Operasi Voltage	5V
Input Voltage	7-12 V (Rekomendasi)
Input Voltage	6-20 V (limits)
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 Mhz

### 2.3 Bluetooth

Bluetooth adalah teknologi Komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping reciever* yang mampu menyediakan layanan Komunikasi data dan suara secara *real time* antara *host-host Bluetooth*. Pada dasarnya *Bluetooth* diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel di dalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi *mobile wireless* dengan biaya yang relatif rendah, konsumsi daya yang rendah, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam-macam.

HC-05 adalah sebuah modul Bluetooth SPP (Serial Port Protokol) yang mudah digunakan untuk Komunikasi serial *wireless* (nirkabel) yang mengkonversi port serial ke Bluetooth. HC-05 menggunakan modulasi Bluetooth V2.0 + EDR (*Enhanced Data Rate*) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz. Modul ini dapat digunakan sebagai *Slave* maupun *Master*. HC-05 memiliki 2 mode konfigurasi, yaitu AT mode dan *Communication mode*. AT mode berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Modul Bluetooth HC-05 ditunjukkan pada Gambar 2.3.



**Gambar 2.3** Bluetooth HC-05

Adapun spesifikasi dari HC-05 adalah :

**1. Hardware :**

1. Sensitivitas -80dBm (Typical).
2. Daya *transmit* RF sampai dengan +4dBm.
3. Operasi daya rendah 1,8V – 3,6V I/O.
4. Kontrol GPIO.
5. Antarmuka UART dengan baudrate yang dapat diprogram.
6. Antenna terintegrasi.

**2. Software :**

1. Default baudrate 9600, Mendukung baudrate : 9600, 19200, 38400, 57600, 115200, 230400 dan 460800.
2. Auto koneksi pada saat *device* dinyalakan (default).
3. Auto reconnect pada menit ke 30 ketika hubungan putus karena range koneksi.

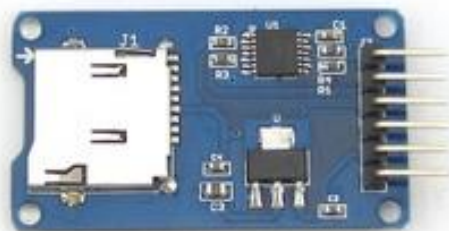
## **2.4 Media Penyimpanan**

Media penyimpanan dalam perekam data berbasis Mikrokontroler berupa media penyimpanan digital. *Secure Digital Card* (SD card) merupakan salah satu media penyimpanan yang banyak digunakan. Saat ini media penyimpanan ini digunakan pada berbagai macam perangkat, seperti kamera digital, telepon genggam, *Personal Digital* 2GB untuk generasi awal dan 4GB hingga 32GB pada generasi akhir yang disebut SDHC card. Pengembangan lebih lanjut dari media penyimpanan ini menghasilkan dimensi yang lebih kecil dan

kompak seiring dengan perkembangan jaman yang berupa Mini SD dan Micro SD.

Layaknya kartu flash lainnya, MicroSD sudah terformat dengan sistem file sebagai FAT16, SDHC sebagai FAT32, sedangkan SDXC sebagai ExFAT. Di manapun FAT16 dan FAT32 memungkinkan untuk dapat diakses melalui semua perangkat host pembaca SD.

Salah satu Media Penyimpanan yang digunakan dengan Mikrokontroler adalah Catalex SD Card. Modul Micro SD Card Adapter adalah modul pembaca kartu Micro SD, melalui sistem file dan SPI antarmuka driver, MCU untuk melengkapi sistem file untuk membaca dan menulis kartu MicroSD. Modul Catalex SD Card ditunjukkan pada Gambar 2.4.



**Gambar 2.4** Catalex SD Card

Fitur modul adalah sebagai berikut:

1. Mendukung kartu micro SD, kartu SDHC (kartu kecepatan tinggi).
2. Kontrol *Interface* ; 6 pin ( GND, VCC, MISO, MOSI, SCK, CS).
3. 3.3V rangkaian regulator tegangan.
4. Micro SD konektor.

## **2.5 Real Time Clock**

RTC (*Real time clock*) adalah jam elektronik berupa chip yang dapat menghitung waktu (mulai detik hingga tahun) dengan akurat dan menjaga/menyimpan data waktu tersebut secara *realtime*

Ds1307 merupakan salah satu Tipe IC RTC (*Real Time Clock*) berdaya rendah yang menggunakan sistem Biner Code Desimal (BCD). Alamat dan data dikirim secara serial melalui I2C. Jam/tanggal dari RTC (*Real Time Clock*) menyediakan detik, menit, jam, hari, tanggal, bulan, dan tahun. Otomatis penyesuaian jumlah hari dalam satu bulan dan penyesuaian jumlah hari dalam tahun biasa dan tahun kabisat.

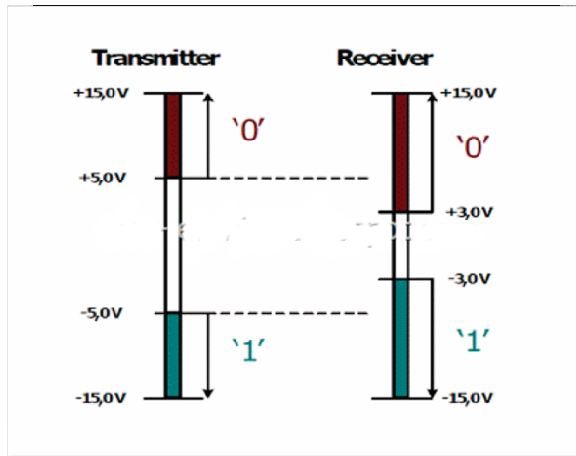


Format jam bisa menggunakan format 24-jam atau 12-jam AM/PM. Ds1307 mempunyai rangkaian yang mendeteksi *power fail* dan secara otomatis mengalihkan daya ke sumber cadangan untuk menyimpan waktu yang tercatat dalam IC. Modul *Real Time Clock* Ds1307 ditunjukan pada Gambar 2.5.

### Gambar 2.5 RTC Ds1307

## 2.6 Serial RS-232

### I. Bentuk Sinyal dan Level Tegangan



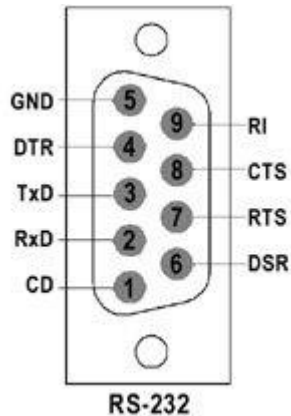
**Gambar 2.6** Level Tegangan RS-232

Penentuan jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki-kaki di konektor. Beberapa parameter yang ditetapkan EIA (*Electronics Industry Association*) antara lain:

1. Sebuah 'spasi' (logika 0) antara tegangan +3 s/d +15 volt
2. Sebuah 'tanda' (logika 1) antara tegangan -3 s/d -15 volt
3. Daerah tegangan antara +3 s/d -3 volt tidak didefinisikan
4. Tegangan rangkaian terbuka tidak boleh lebih dari 25 volt (dengan acuan ground)
5. Arus hubung singkat rangkaian tidak boleh lebih dari 500 mA.

## II. Konektor

standard RS-232 menentukan pula jenis-jenis sinyal yang dipakai mengatur pertukaran informasi antara DTE dan DCE, semuanya terdapat 24 jenis sinyal tapi yang umum dipakai hanyalah 9 jenis sinyal. Konektor yang dipakai pun ditentukan dalam standard RS-232, untuk sinyal yang lengkap dipakai konektor DB25, sedangkan konektor DB9 hanya bisa dipakai untuk 9 sinyal yang umum dipakai. Pin Konektor DB9 ditunjukkan pada Gambar 2.7.



**Gambar 2.7** Konektor DB9

**Tabel 2.3** Fungsi Pin DB9

Singkatan	Keterangan	Fungsi
TD	<i>Transmit Data</i>	Untuk pengiriman data serial
RX	<i>Receive Data</i>	Untuk penerimaan data serial
RTS	<i>Request To Send</i>	Sinyal untuk menginformasikan perangkat UART telah siap
CTS	<i>Clear To Send</i>	Digunakan untuk memberitahukan bahwa perangkat siap untuk melakukan pertukaran data
DSR	<i>Data Set Ready</i>	Memberitahukan UART bahwa perangkat siap untuk melakukan pertukaran data
SG	<i>Signal Ground</i>	Dihubungkan ke ground
CD	<i>Carrier Detect</i>	Saat perangkat mendeteksi suatu <i>carrier</i> , dari perangkat lain, maka sinyal ini akan diaktifkan
DTR	<i>Data Terminal Ready</i>	Untuk memberitahukan bahwa uart siap melakukan pertukaran data
RI	<i>Ring Indikator</i>	Akan aktif jika ada sinyal masuk

Tabel 2.3 menjelaskan fungsi pin masing-masing Konektor DB9. Jika Peralatan yang digunakan menggunakan Level tegangan TTL, Maka Level tegangan RS-232 harus dikonversikan terlebih dahulu ke Level tegangan TTL. Untuk menghubungkan keduanya agar dapat berkomunikasi, dibutuhkanlah sebuah konverter. Kebutuhan sebuah

konverter yang dapat berfungsi dua arah sekaligus, yaitu RS-232 ke TTL dan TTL ke RS-232, dapat menggunakan sebuah IC MAX-232. Serial Converter MAX-232 ditunjukkan pada Gambar 2.8.



**Gambar 2.8** Serial RS-232

## 2.7 Komunikasi Wireless

Komunikasi *wireless* adalah suatu operasi tanpa menggunakan suatu media yang terlindung atau terbungkus seperti kabel, tetapi menggunakan media udara sebagai jalur Komunikasi untuk mengirimkan sinyal atau data pada setiap tujuannya. Sistem *wireless* menggunakan suatu gelombang radio atau gelombang elektromagnetik sebagai jalur Komunikasinya

Keuntungan menggunakan Komunikasi *Wireless* :

1. Untuk rentang jarak di luar kemampuan kabel biasa,
2. Untuk menyediakan link Komunikasi cadangan jika terjadi kegagalan jaringan normal,
3. Untuk menghubungkan komputer portabel atau sementara,
4. Untuk mengatasi situasi di mana pemasangan kabel normal adalah sulit atau finansial tidak praktis, atau
5. Untuk jarak jauh menghubungkan pengguna ponsel atau jaringan.

Komunikasi *Wireless* dapat melalui :

1. Frekuensi radio Komunikasi.
2. Komunikasi gelombang mikro, misalnya jangka panjang line-of-sight melalui antenna.
3. Inframerah (IR) Komunikasi jarak pendek, misalnya dari remote kontrol atau melalui Inframerah.

Salah satu *Device* yang digunakan untuk Komunikasi *Wireless* yaitu NRF24L01+. NRF24L01+ adalah sebuah modul Komunikasi yang memanfaatkan gelombang RF 2,4 GHz ISM ( Industrial, Scientific and Medical). Modul ini menggunakan antar muka SPI ( Serial Peripheral Interface) untuk berkomunikasi. NRF24L01+ mengintegrasikan pengirim lengkap 2,4 GHz RF, RF pengumpul, dan akselerator protokol berupa *Enhanced Shockbrust* yang mendukung antarmuka SPI kecepatan tinggi untuk Kontroler aplikasi. NRF24L01+ memiliki solusi terkait daya ultra rendah yang memungkinkan daya tahan baterai berbulan-bulan. Modul *Transceiver* NRF24L01+ ditunjukkan pada Gambar 2.9.



**Gambar 2.9** Modul NRF24L01+

Modul nirkabel NRF24L01+ memiliki 8 buah pin, diantaranya : VCC (3,3VDC), GND, CE, CSN, MOSI, MISO, SCK dan IRQ. Modul ini memiliki beberapa fitur antara lain :

1. Beroperasi pada ISM 2.4 GHz.
2. Kecepatan pengiriman data 250 kbps hingga 2 Mbps.
3. Operasi daya rendah.
4. Penanganan paket data otomatis.
5. Penanganan transaksi paket otomatis.
6. Sumber daya hanya 1.9 V sampai 3.6 V.
7. Jangkauan pengiriman : 300 meter di tempat terbuka.

## **2.8 Wireless Sensor Network (WSN)**

### **2.8.1 Pengertian dan Perkembangan WSN**

*Wireless Sensor Network* atau jaringan sensor nirkabel adalah kumpulan sejumlah node yang diatur dalam sebuah jaringan kerjasama.. Masing-masing node dalam jaringan sensor nirkabel biasanya dilengkapi dengan radio *transceiver* atau alat Komunikasi *wireless* lainnya, mikrokontroler, dan sumber energi seperti baterai. Banyak aplikasi yang bisa dilakukan menggunakan jaringan sensor nirkabel, misalnya pengumpulan data kondisi lingkungan.

Perkembangan teknologi semakin mengarah kepada konektivitas lingkungan fisik. Kebanyakan observasi yang dilakukan di lapangan melibatkan banyak faktor dan parameter-parameter untuk mendapatkan hasil yang maksimal dan akurat. Jika peneliti hendak mengambil informasi langsung di lapangan, maka kendalanya adalah dibutuhkan biaya yang besar dan waktu yang lama untuk mendeteksi fenomena yang muncul sehingga menyebabkan kemampuan yang tidak efisien dan tidak praktis. Dengan adanya teknologi WSN, memungkinkan peneliti untuk mendapat informasi yang maksimal tanpa harus berada di area sensor. Informasi dapat diakses dari jarak jauh melalui gadget seperti laptop, remote *device*, server dan sebagainya.

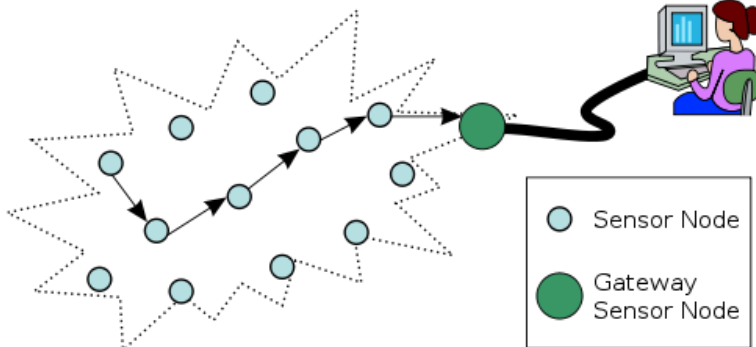
Berikut adalah beberapa keuntungan yang bisa diperoleh dari teknologi WSN.

1. Praktis karena tidak perlu ada instalasi kabel yang rumit dan dalam kondisi geografi tertentu sangat menguntungkan dibanding *wired* sensor.
2. Sensor menjadi bersifat *mobile*, artinya pada suatu saat dimungkinkan untuk memindahkan sensor untuk mendapat pengukuran yang lebih tepat tanpa harus khawatir mengubah desain ruangan maupun susunan kabel ruangan.
3. Meningkatkan efisiensi secara operasional.
4. Mengurangi total biaya sistem secara signifikan.
5. Dapat mengumpulkan data dalam jumlah besar.
6. Konfigurasi *software* lebih mudah.
7. Memungkinkan Komunikasi digital 2 arah.

### **2.8.2 Arsitektur WSN**

Pada WSN, node sensor disebar dengan tujuan menangkap adanya gejala atau fenomena yang hendak diteliti. Jumlah node yang disebar dapat ditentukan sesuai kebutuhan dan tergantung beberapa faktor

misalnya luas area, kemampuan sensing node, dan sebagainya. Tiap node dalam WSN dapat melakukan pemantauan lingkungan terbuka secara langsung dengan memanfaatkan beberapa macam sensor. Arsitektur WSN secara umum dapat ditunjukkan pada Gambar 2.10.



**Gambar 2.10** Arsitektur WSN

### 2.8.3 Bagian-Bagian WSN

1. *Transceiver*, berfungsi untuk menerima/mengirim data dengan menggunakan protokol IEEE 802.15.4 atau IEEE 802.11b/g kepada *device* lain.
2. Mikrokontroler, berfungsi untuk melakukan fungsi perhitungan, mengontrol dan memproses *device* yang terhubung dengan mikrokontroler.
3. *Power source*, berfungsi sebagai sumber energi bagi sistem *wireless* sensor.
4. External memory, berfungsi sebagai tambahan memory bagi sistem *wireless* sensor, pada dasarnya sebuah unit Mikrokontroler memiliki unit memory sendiri.
5. Sensor, berfungsi untuk men-sensing besaran-besaran fisis yang hendak diukur. Sensor adalah suatu alat yang mampu untuk mengubah suatu bentuk energi ke bentuk energi lain, dalam hal ini adalah mengubah dari energi besaran yang diukur menjadi energi listrik yang kemudian diubah oleh ADC menjadi deretan pulsa terkuantisasi yang kemudian bisa dibaca oleh Mikrokontroler.

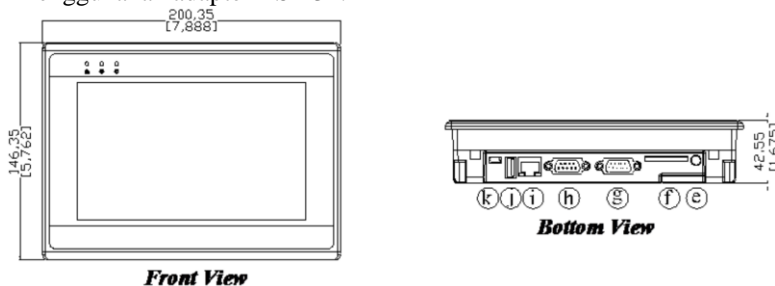
## 2.9 HMI (*Human Machine Interface*)

*Human Machine Interface* adalah unit kontrol terpusat yang dilengkapi dengan penerimaan data, event logging, video feed, dan pemicu. HMI dapat digunakan untuk mengakses sistem setiap saat untuk berbagai tujuan, misalnya untuk menampilkan kesalahan mesin, menampilkan kesalahan status proses, menampilkan jumlah produk, dimana operator melakukan pengendalian mesin. Penggunaan HMI memiliki beberapa keuntungan, misalnya penggunaan kode warna sehingga mudah dikenali, dan layar yang dapat dirubah-rubah. Pada Tugas Akhir ini HMI harus bekerja secara terintegrasi dengan Mikrokontroler ATmega 2560. Mikrokontroler ATmega 2560 akan mengambil informasi data sensor yang dikirim dari masing-masing SLAVE dan juga bisa melakukan pengendalian ke Setiap SLAVE.

### 2.9.1 WienView Weintek MT8070iH

WienView Weintek MT8070iH merupakan salah satu piranti yang dapat digunakan Sebagai HMI (*Human Machine Interface*)., berupa layar sentuh (*touchscreen*). Tampilan dari Weintek MT8070iH dibuat dengan bantuan *Software Easy Builder 8000* pada sebuah personal computer (PC).

Pembuatan program pada tampilan Weintek MT8070iH disusun berdasar pengalaman program yang terdapat pada Mikrokontroler ATmega 2560. Komunikasi hubungan Weintek MT8070iH dengan Mikrokontroler ATmega 2560, merupakan hubungan antarmuka menggunakan adapter RS-232.



**Gambar 2.11** Dimensi MT8070iH



WienView Weintek MT8070iH dapat menampilkan elemen-elemen dengan banyak variasi seperti karakter, angka, lampu, touch switch dan graph pada layar. Gambar 2.11 menunjukkan contoh dimensi layar WienView Weintek MT8070iH.



**Gambar 2.12** Port Weintek MT8070iH

Pada Gambar 2.12. menunjukan port yang terdapat pada Weintek MT8070iH. Ukuran pixel pada WienView Weintek MT8070iH dihitung dari besarnya layar yaitu 800 X 480 pixel. Berikut ini merupakan beberapa Fasilitas yang terdapat pada WienView Weintek MT8070iH.

1. Karakter (text), gambar dan karakter (text) dapat tertulis secara langsung pada layar, maka perlu table memori untuk menampilkannya. Karakter (text) biasanya digunakan sebagai keterangan untuk memudahkan pengguna.
2. Karakter (*character string memory table*), Karakter string disimpan di dalam table memori karakter string yang dapat ditampilkan. Tampilan karakter dapat diubah dengan mengubah data yang disimpan di dalam table memory karakter string.
3. Angka (*numeral memory table*), angka-angka disimpan di dalam table memori angka dan dapat ditampilkan. Tampilan angka dapat diganti dengan mengganti data yang disimpan di dalam table memori angka. Nilai Hexadesimal juga dapat ditampilkan.
4. Lampu, lampu dapat digunakan untuk status indikator status pengoperasian. Tampilan persegi, lingkaran, dan segi banyak digunakan untuk indikator.
5. *Touch Switches*, *Touch switch* dapat diatur dimanapun pada layar. Menekan tombol pada layar mempunyai beberapa fungsi antara lain : pemberitahuan bahwa tombol telah ditekan. Masukan angka atau karakter string (fungsi tombol masukan), menyalin angka atau karakter string (fungsi tombol penyalin) dan lain sebagainya.
6. *Graphs*, *Bar Graps* dapat ditampilkan menurut angka yang disimpan pada table memori angka.

## 2.10 Protokol Modbus

Modbus adalah protokol Komunikasi serial yang dipublikasikan oleh MODICON pada tahun 1979 untuk diaplikasikan ke dalam Programable Logic Controller (PLC). Modbus sudah menjadi standar protokol yang umum digunakan untuk menghubungkan peralatan elektronik industri. Beberapa alasan mengapa protokol ini banyak digunakan, antara lain :

1. Modbus dipublikasikan secara terbuka dan bebas royalti.
2. Mudah digunakan dan dipelihara.
3. Memindahkan data bit atau word tanpa terlalu banyak membatasi vendor.

Modbus mampu menghubungkan 247 peralatan (Slave) dalam satu jaringan atau Master, misalnya sebuah sistem yang melakukan pengukuran suhu dan kelembaban dan mengirimkan hasilnya ke sebuah computer. Modbus sering digunakan untuk menghubungkan computer pemantau dengan remote terminal unit (RTU) pada sistem *supervisory control and data acquisition* (SCADA).

Dalam jenisnya Modbus dapat dikategorikan dalam varian sebagai berikut :

1. Modbus RTU : merupakan varian Modbus yang ringkas dan digunakan pada Komunikasi serial. Format RTU dilengkapi dengan mekanisme *cyclic redundancy error* (CRC) untuk memastikan keandalan data. Modbus RTU merupakan implementasi protokol Modbus yang paling umum digunakan. Setiap frame data dipisahkan dengan periode idle (*silent*).
2. Modbus ASCII : digunakan pada Komunikasi serial dengan memanfaatkan karakter ASCII. Format ASCII menggunakan mekanisme *longitudinal redundancy check* (LRC). Setiap frame data Modbus ASCII diawali dengan titik dua (":") dan baris baru yang mengikuti (CR/LF).
3. Modbus TCP/IP atau Modbus TCP : merupakan varian Modbus yang digunakan pada jaringan TCP/IP.

Variasi Modbus dapat diaplikasikan pada port serial dan Ethernet dan jaringan lainnya yang support dengan internet protokol suite. Sebagian besar peralatan Modbus menggunakan port serial RS-232 maupun RS-485. Konsep dasar Komunikasi Modbus terdiri dari Master dan Slave. Sebuah perintah Modbus dilengkapi dengan alamat tujuan

perintah tersebut. Hanya alamat tujuan yang akan memproses perintah, meskipun peralatan yang lain mungkin menerima perintah tersebut. Setiap perintah Modbus memiliki informasi pemeriksaan kesalahan untuk memastikan data diterima tanpa kerusakan. Perintah dasar Modbus RTU dapat memerintahkan peralatan untuk mengubah nilai registernya, mengendalikan dan membaca port I/O, serta memerintahkan peralatan untuk mengirimkan kembali nilai yang ada pada registernya.

Pada protokol modbus terdapat 4 buah jenis penyimpanan data dengan panjang masing-masing 16 bit. Register yang terdapat pada Modbus ditunjukkan pada Tabel 2.4.

**Tabel 2.4** Register Protokol Modbus

<b>Primary tables</b>	<b>Object type</b>	<b>Type of</b>	<b>Comments</b>
Coils	Single bit	Read-Write	Master dan Slave bisa merubah data coil
Discretes input	Single bit	Read-Only	Data hanya bisa di rubah oleh Slave
Input Registers	16-bit word	Read-Only	Data hanya bisa di rubah oleh Slave
Holding Registers	16-bit word	Read-Write	Master dan Slave bisa merubah data register

#### 1. Coil

Pada mulanya jenis data ini digunakan untuk mengaktifkan coil relay . nilai jenis data ini ON atau OFF. Coil mempunyai panjang 16 bit, sehingga untuk mengaktifkan/ON dgn cara memberi nilai FF00H dan 0000H untuk OFF. data FF00 dan 00 disimpan di register 00000 sampai 09999.

#### 2. Input Relay / input biner / input digital/input diskrit

kebalikan dengan coil, input relay digunakan untuk mengetahui status relay apakah sedang ON atau OFF. Input relay bersifat read only bagi Master dan hanya bisa dirubah oleh Slave saja. data tsb disimpan di register 10001 sampai 19999.

#### 3. Input Register

Input Regsiter digunakan untuk menyimpan data analog dgn range nilai 0 ~ 65535 . Input register bersifat read only bagi Master. data ini disimpan di register ber nomor 30001 sampai 39999.

#### 4. Holding Register

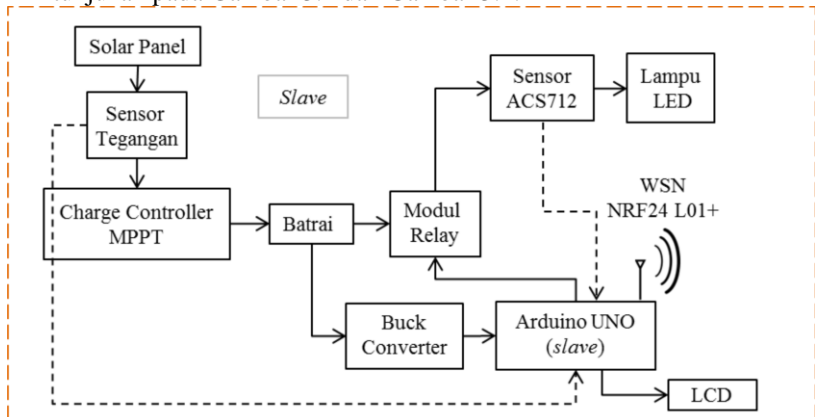
Holding Register digunakan untuk menyimpan nilai dengan range 0~65535 .register ini mempunyai alamat register 40001 sampai 49999.

-----Halaman ini sengaja dikosongkan-----

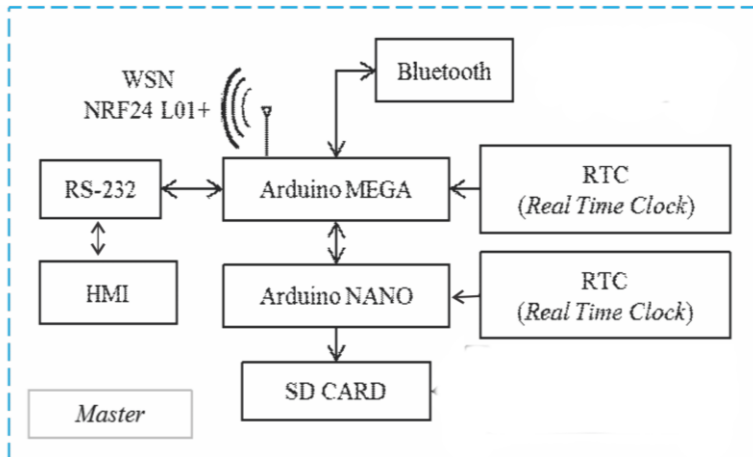
### BAB III

## PERANCANGAN SISTEM KONTROL

Dalam perancangan Tugas Akhir Rancang Bangun Pengendali Utama pada Sistem Kontrol dan Monitoring Lampu Penerangan Jalan Umum Tenaga Surya menggunakan *Wireless Sensor Network* dengan *Human Machine Interface* Terpusat, Pembuatan Tugas Akhir ini dibagi menjadi 2 bagian yaitu Master dan Slave. “Master”, pada bagian ini digunakan untuk Melakukan Perintah Kontrol dan Monitoring Masing-masing Lampu PJU-TS. Terdapat 2 mode perintah kontrol yang terdapat pada Master, yaitu Mode Manual dan Otomatis. Mode pertama adalah mode manual, operator dapat menyalakan/mematikan lampu sesuai kebutuhan penerangan atau Kondisi PJU-TS. Mode kedua adalah mode penjadwalan, mode ini menggunakan RTC (*Real Time Clock*), secara otomatis lampu (on/off) sesuai jadwal. Sedangkan “Slave” digunakan untuk Mengukur Besarnya Arus dan Tegangan Kemudian Mengirimkannya ke bagian “Master”. Berikut ini merupakan blok fungsional diagram masing-masing bagian sistem secara keseluruhan. Ditunjukkan pada Gambar 3.1 dan Gambar 3.2.



**Gambar 3.1** Slave Sistem Lampu PJU-TS



**Gambar 3.2** Master Sistem Lampu PJU-TS

Sesuai dengan blok diagram diatas memiliki dua bagian utama yakni Slave dan Master. Berikut pembagian masing-masing tugas, yang akan dibahas pada Bab ini.

1. Nurul Qomariya : Slave
2. Raka Satria Pradana H.D. : Master



**Gambar 3.3** Ilustrasi Proses Pengiriman Data

Metode Komunikasi yang digunakan dapat diilustrasikan seperti Gambar 3.3 penjelasannya sebagai berikut :

1. Proses Kontrol dan Monitoring Tegangan Solar Cell dan Arus Lampu PJU-TS di lakukan di bagian Master dengan *interface* menggunakan *Human Machine Interface* WienView MT8070iH dan Android.
2. Masing-masing Slave memiliki interval waktu untuk mengirimkan data menuju ke bagian Master, data yang dikirim setiap slave

berupa variable nilai Tegangan Solar Cell dan Arus dari Lampu PJU-TS. Sebelum data tersebut dikirim oleh Slave, data tersebut dibentuk kedalam paket data dengan format, ID Slave Pengirim, Data Tegangan Solar cell dan Arus Lampu PJU-TS.

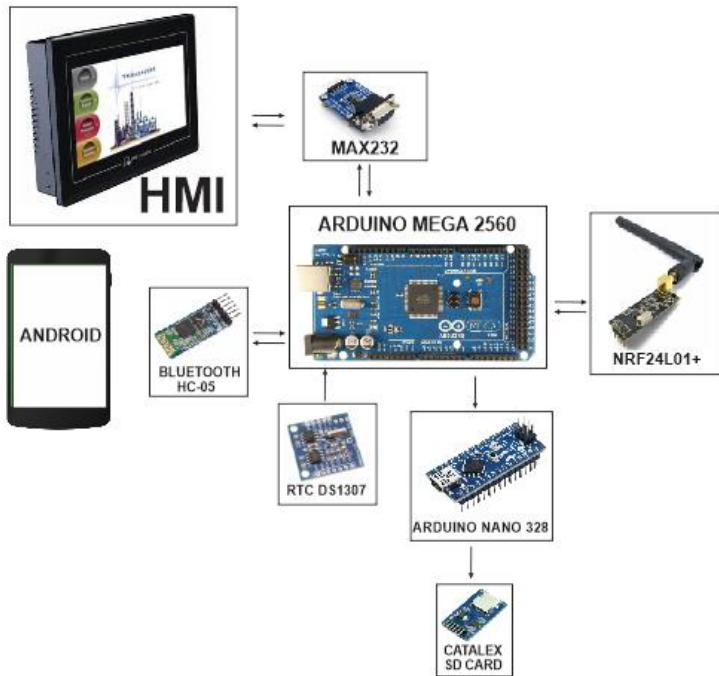
3. Master hanya berkomunikasi dengan Slave 1, Jika Master ingin melakukan aktifitas Kontrol maka perintah tersebut akan dikirimkan ke bagian Slave 1. lalu di cek apakah perintah tersebut untuk Slave 1 atau tidak, jika tidak maka perintah tersebut akan di teruskan ke Slave selanjutnya sampai perintah tersebut sampai ke Slave tujuan.
4. Begitu juga Perintah Kirim data dari Slave ke Master. Slave akan mengirimkan data sesuai interval yang telah ditentukan. Sebelum Data dikirim, data akan dirubah menjadi Paket data yang berisi ID pengirim, Tegangan Solar Cell, dan Arus Lampu PJU-TS. Data lalu di kirim ke Slave sebelumnya, Kemudian di Slave tersebut akan di cek apakah perintah Kontrol atau kirim data, apabila perintah kirim data dan penerima tersebut bukan Master maka Data akan di teruskan sampai menuju ke bagian Master.
5. Setelah data Sampai di bagian Master, data tersebut kemudian dilakukan parsing, gunanya untuk memecah data sesuai dengan ID pengirim dan nilai dari Data Tersebut (Data Tegangan Solar Cell dan Arus Lampu PJU-TS).

Selanjutnya pada BAB ini akan dibahas mengenai 2 Sub BAB bagian dari Master yaitu :

1. Perancangan Perangkat Keras
2. Perancangan Perangkat Lunak

### **3.1 Perancangan Perangkat Keras**

Pada bab ini berisi bagaimana tahapan yang dilakukan dalam perencanaan dan pembuatan Tugas Akhir Pengendali utama Sistem Lampu PJU-TS. Penjelasan diawali dengan penjelasan blok fungsional sistem secara keseluruhan, kemudian perancangan perangkat keras dan diakhiri dengan perangkat lunak. Adapun perancangan perangkat keras dilakukan pada perancangan Keras bagian Master pada sistem Kontrol dan Monitoring Lampu PJU-TS. Sedangkan perancangan perangkat lunak berisi tentang algoritma yang terdapat pada Master dan *Design Human Machine Interface* menggunakan WienView MT8070iH. Secara keseluruhan rancangan Sistem PJU-TS dapat dilihat pada Gambar 3.4.



**Gambar 3.4** Diagram Bagian Master

Bagian Master digunakan untuk mengendalikan dan memonitoring kondisi dari masing-masing lampu PJU-TS apakah dalam kondisi baik atau tidak. objek yang di Monitoring berupa Tegangan dari Solar Cell dan Arus dari lampu PJU. Data hasil pembacaan Arus dan tegangan tersebut kemudian dikirim ke Master untuk di Monitoring oleh operator, dan juga data tersebut disimpan dalam SD Card yang berguna untuk pencatatan kondisi masing-masing lampu PJU-TS. Selain itu dibagian Master operator bisa melakukan aktifitas kendali (Kontrol) lampu PJU-TS secara manual atau otomatis. Bagian-bagian dari blok diagram sistem kami yaitu :

- |                           |                            |
|---------------------------|----------------------------|
| 1. Kontroler Utama        | : Arduino Mega ATmega 2560 |
| 2. Kontroler Kedua        | : Arduino Nano ATmega 328  |
| 3. Komunikasi antar Slave | : SPI (NRF24L01)           |



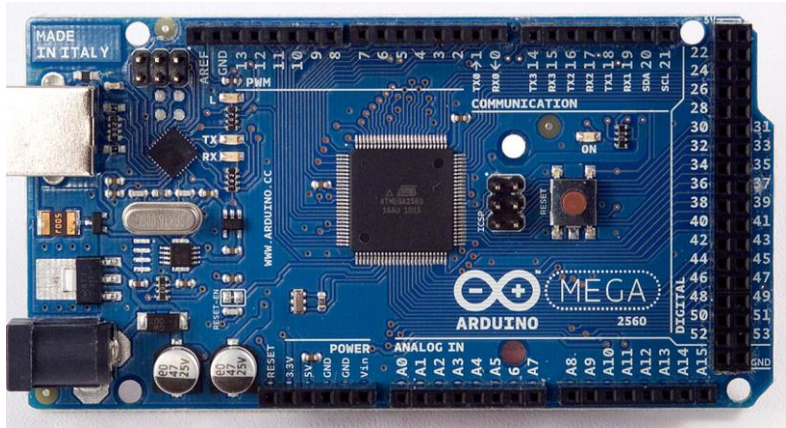
4. Komunikasi Kontroler Kedua : Serial UART
5. *Interface* : Android (*App-Inventor*) & Wienview
6. Pencatatan Data : Catalex SD Card Module

Dalam perancangan Perangkat Rancang Bangun Pengendali Utama pada Sistem Kontrol dan Monitoring Lampu PJU-TS menggunakan *wireless sensor network* dengan HMI terpusat diperlukan 2 Kontroler yaitu arduino Mega dan arduino Nano. Arduino Mega digunakan sebagai Kontroler utama yang bertugas untuk melakukan perintah kontrol ke masing-masing Slave dan Menerima data tegangan *Solar Cell* dan Arus dari Lampu PJU-TS. Data tersebut diterima dari masing-masing Slave yang berada di lampu PJU-TS. Sedangkan Arduino Nano digunakan untuk menyimpan Data Arus dan Tegangan tersebut ke SD Card. Digunakan 2 Kontroler karena fungsi pin SPI (*Serial Peripheral Interface*) dari Arduino Mega digunakan oleh *Device Transreceiver* NRF24L01+ Untuk berkomunikasi antar Slave oleh sebab itu digunakan pin SPI Arduino Nano untuk Datalogger ke SD Card, dengan cara mengirim data yang diterima Arduino Mega Kemudian mengirimkannya lagi ke Arduino Nano melalui Serial (UART). Di Master juga terdapat RTC (*Real Time Clock*) yang digunakan untuk mengambil data Waktu untuk menjalankan fungsi Otomatis Sistem Kontrol Lampu PJU-TS. Selain itu di Bagian Master Juga Terdapat Bluetooth HC-05 yang digunakan untuk berkomunikasi dengan smartphone Android sebagai *Interface* baik Kontrol & Monitoring oleh operator.

### **3.1.1 Rangkaian Mikrokontroler ATMega 2560**

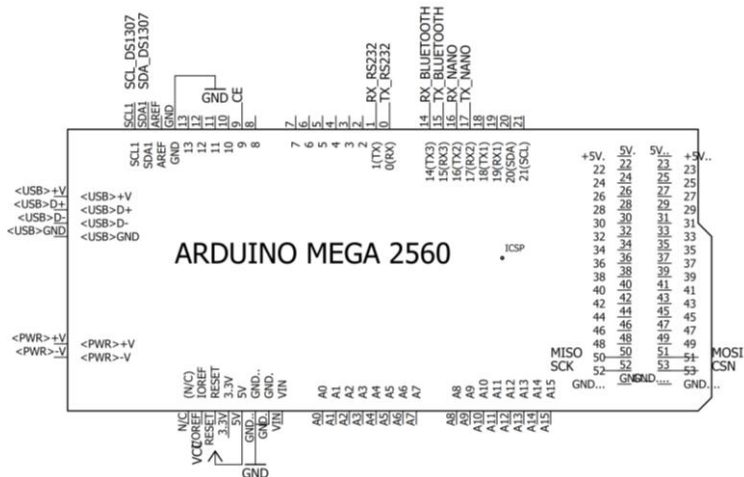
Arduino MEGA 2560 digunakan sebagai kontroler utama dari Master. Arduino Mega memiliki 2 fungsi, yaitu Perintah Kontrol dan Monitoring. Terdapat 2 mode dalam melakukan perintah Kontrol yaitu mode Manual dengan Otomatis. Mode Manual yaitu digunakan untuk mengendalikan lampu sesuai keinginan operator bisa karena dilihat dari kondisi Monitoring lampu PJU-TS tersebut, Sedangkan Mode Otomatis yaitu digunakan untuk mengendalikan lampu PJU-TS sesuai dengan jadwal yang telah ditentukan. Perintah kontrol dan Monitoring ini bisa dilakukan dengan menggunakan Android. Fungsi yang kedua adalah Monitoring adalah menerima data Tegangan dari Solar Cell dan Arus Lampu PJU-TS yang dikirim dari masing-masing Slave, kemudian akan

di kirim secara *wireless* menggunakan Bluetooth ke Android sebagai *Interfacenya*, selain itu data tersebut juga di simpan di dalam SD Card yang berguna untuk pencatatan kondisi masing-masing lampu PJU-TS.. Gambar 3.5 adalah gambar board Arduino Mega 2560.



**Gambar 3.5** Rangkaian Arduino Mega 2560

Berikut ini adalah diagram fungsional dari keseluruhan alat seperti pada Gambar 3.6.



**Gambar 3.6** Fungsi Pin Arduino Mega

Berikut penjelasan penggunaan masing-masing Pin Arduino Mega 2560 dengan *device* lain pada Bagian Master dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Fungsi Pin Arduino Mega 2560

No.	Pin Arduino Mega	Keterangan
1.	Serial.0	Komunikasi dengan HMI MT8070iH
2.	Serial.2	Kirim data ke Arduino Nano328
3.	Serial. 3	Komunikasi dengan Bluetooth HC-05
4.	SPI Mega2560	Komunikasi dengan NRF24L01+
5.	I2c Mega2560	Komunikasi dengan RTC Ds1307
6.	A6	Indikator Terima
7.	A7	Indikator Kirim

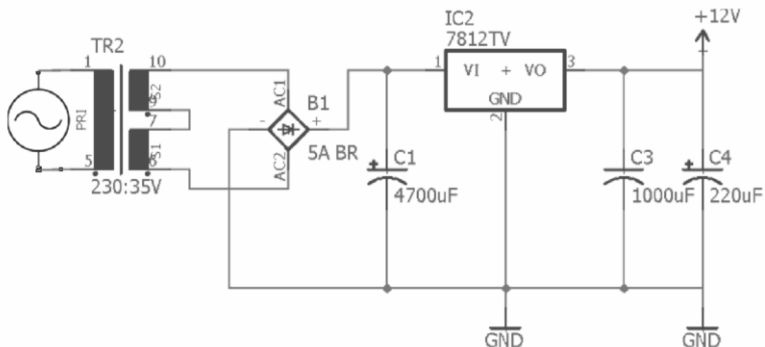
### 3.1.2 Rangkaian *Power Supply*

*Power Supply* digunakan untuk menyediakan tegangan DC untuk keseluruhan kebutuhan Daya bagi Perangkat keras Sistem Lampu PJU-TS. Secara prinsip rangkaian *power supply* adalah menurunkan tegangan AC lalu menyearahkan tegangan AC sehingga menjadi tegangan DC, yang terdiri atas transformator, dioda dan kapasitor/kondensator.

Dalam perancangan *power supply*, sebaiknya *power supply* tersebut dapat menghasilkan tegangan DC murni, berikut hal-hal yang perlu diperhatikan dalam perancangan *power supply* :

1. Transformator Step Down berfungsi untuk menurunkan Tegangan AC yang bersumber dari tegangan 220 Volt AC. Tegangan tersebut diturunkan agar sesuai dengan *supply* yang dibutuhkan oleh Mikrokontroler.
2. Dioda Penyearah atau diode bridge berfungsi untuk menyearahkan tegangan AC menjadi tegangan DC yang dibutuhkan oleh perangkat keras Mikrokontroler.
3. Kapasitor yang berfungsi sebagai filter atau penyaring untuk mengurangi ripple dan menghasilkan DC murni dan sumber yang stabil bagi Mikrokontoler.
4. Regulator yang berfungsi sebagai regulator tegangan/menurunkan Tegangan yang dibutuhkan oleh Mikrokontoler.

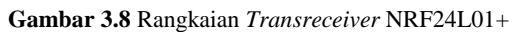
Berikut dapat dilihat rangkaian *power supply* yang penulis rancang pada Gambar 3.7.

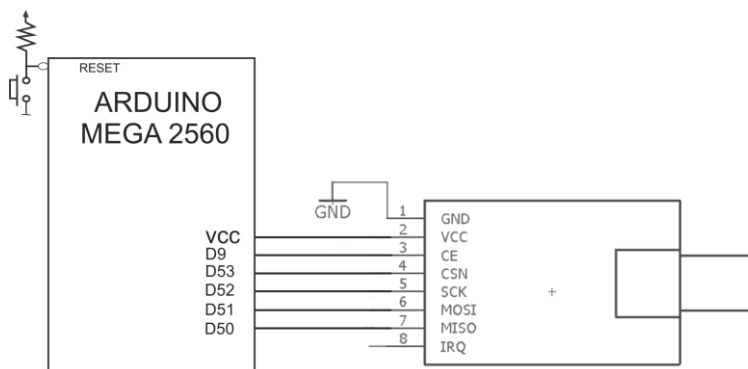


**Gambar 3.7** Rangkaian *Power Supply*

### 3.1.3 Rangkaian NRF24L01+

Modul *Transceiver* NRF24L01+ ini menggunakan antarmuka SPI (*Serial Peripheral Interface*) untuk Komunikasi *Wireless* Antara Master dengan sensor. *Serial Peripheral Interface* (SPI) merupakan salah satu mode komunikasi serial synchronous kecepatan tinggi yang dimiliki oleh Mikrokontroler. Komunikasi SPI membutuhkan 3 jalur yaitu MOSI, MISO, dan SCK. Melalui komunikasi ini data dapat saling dikirimkan baik antara mikrokontroler maupun antara mikrokontroler dengan peripheral lain di luar mikrokontroler. Walaupun NRF24L01+ memiliki kemampuan *transmitter* dan *receiver* tetapi hanya bisa digunakan secara satu arah (*half duplex*) jadi fungsi Transmitter dan Receiver tidak digunakan secara bersamaan. Untuk berkomunikasi NRF24L01+ menggunakan pipe sebagai saluran data sebesar 5 byte dalam bilangan hexadecimal. Modul ini juga mempunyai kecepatan pengiriman data (datarate) 250kbps, 1Mbps dan 2Mbps selain itu terdapat pengaturan power amplifier untuk penggunaan daya dan jarak modul ini. Modul NRF24L01+ yang berada di bagian Master digunakan untuk berkomunikasi dengan 3 buah Slave lampu PJU-TS, baik kirim data (Kontrol) maupun terima data (Monitoring). Sebelum Data dikirim oleh Slave, data Tegangan dan Arus tersebut dibuat dalam suatu paket data kemudian baru dikirim, data tersebut dikirim dalam bentuk String (kata). Setelah data tersebut diterima di Master, Data tersebut kemudian





**Gambar 3.9** Wiring Pin Transceiver NRF24L01+

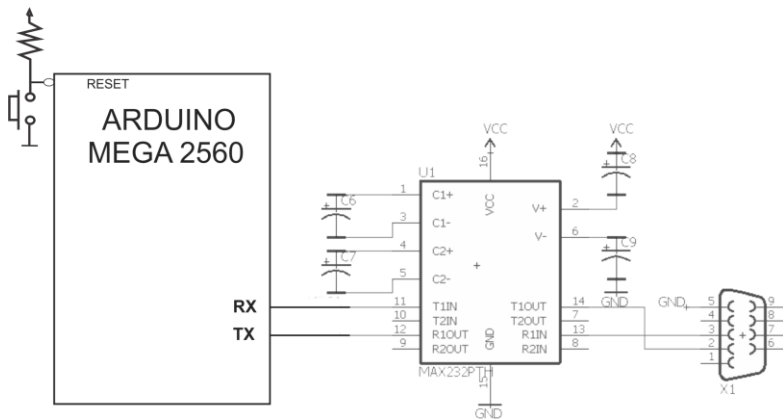
Berikut penjelasan penggunaan masing-masing Pin pada NRF24L01+ dengan Arduino Mega 2560 dapat dilihat pada Tabel 3.2.

**Tabel 3.2** Wiring NRF24L01+ dengan Arduino Mega 2560

No.	Pin NRF24L01+	Pin Arduino Mega 2560
1.	CE	Pin Digital No. 9
2.	CSN	Pin Digital No. 53
3.	SCK	Pin Digital No. 52
4.	MOSI	Pin Digital No. 51
5.	MISO	Pin Digital No. 50
6.	IRQ	

### 3.1.4 Rangkaian Serial RS-232

RS-232 adalah standar Komunikasi serial yang digunakan untuk koneksi peripheral ke peripheral. Pada perancangan ini RS-232 digunakan untuk koneksi antara Arduino Mega dan HMI (*Human Machine Interface*). Untuk Menyamakan Level Tegangan TTL Mikrokontroler ATmega agar sesuai dengan HMI (*Human Machine Interface*) Weintek diperlukan sebuah driver IC MAX-232. Pada dasarnya IC ini memerlukan komponen tambahan berupa kapasitor eksternal yang dipasang pada pin-pin tertentu. Kapasitor ini merupakan rangkaian baku yang berfungsi sebagai *charger pump* untuk suplay muatan ke bagian pengubah tegangan. Wiring diagram Converter RS-232 dapat dilihat pada Gambar 3.10 sebagai berikut :



**Gambar 3.10** Rangkaian Serial MAX-232

Berikut penjelasan penggunaan masing-masing Pin pada RTC Serial RS-232 dengan Arduino Mega 2560 dapat dilihat pada Tabel 3.3.

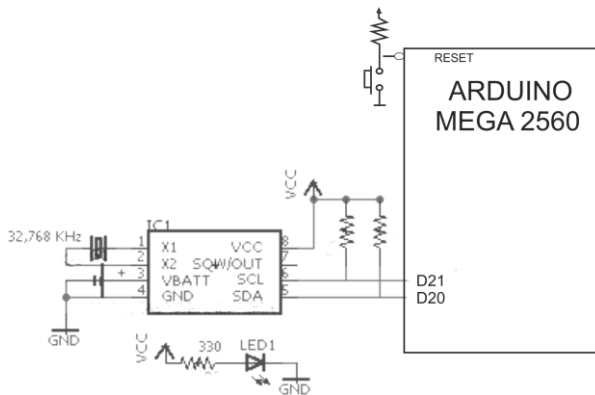
**Tabel 3.3** Wiring RS-232 dengan Arduino Mega 2560

No.	Pin RS-232	Pin Arduino Mega 2560
1.	T1IN	Pin Digital No. 1 (TX)
2.	R1OUT	Pin Digital No. 0 (RX)

### 3.1.5 Rangkaian Real Time Clock Ds1307

RTC (*Real Time Clock*) Ds1307 ini digunakan untuk mendapatkan data Waktu Secara *Realtime*, data yang dikirimkan oleh RTC ini adalah data BCD (*Binary Code Digit*). Komunikasi antara Arduino Nano dengan RTC menggunakan I2c (*Inter Integrated Circuit*) dengan pin SDA (Serial Data), SCL (Serial Clock), Serial Data digunakan untuk mengirimkan data sedangkan Serial Clock digunakan sebagai sumber clock karena I2c karena merupakan jenis komunikasi *asynchronous*.

RTC memiliki 2 fungsi yaitu untuk mengambil data waktu yang digunakan sebagai mode otomatis dan sebagai pencatatan/pewaktu data Sensor Arus dan tegangan. Wiring diagram Real Time Lock Ds1307 dapat dilihat pada Gambar 3.11 sebagai berikut :



**Gambar 3.11** Rangkaian RTC Ds1307

Berikut penjelasan penggunaan masing-masing Pin pada RTC Ds1307 dengan Arduino Mega 2560 dapat dilihat pada Tabel 3.4.

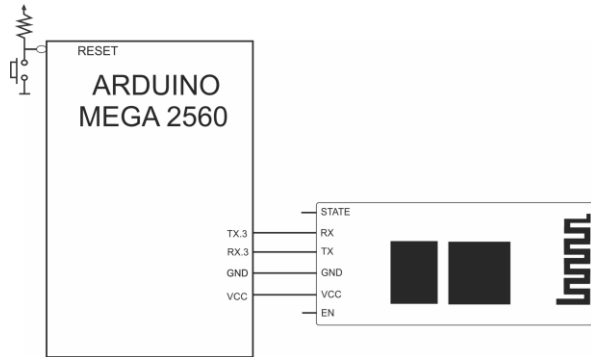
**Tabel 3.4** Wiring Ds1307 dengan Arduino Mega 2560

No.	Pin DS1307	Pin Arduino Mega 2560
1.	SDA	SDA/ Pin Digital No. 20
2.	SCL	SCL/ Pin Digital No. 21

### 3.1.6 Rangkaian Bluetooth HC-05

Rangkaian Bluetooth HC-05 dengan Arduino Mega 2560 ditunjukkan seperti pada Gambar 3.12 digunakan untuk Komunikasi *wireless* ke Android. Rangkaian Bluetooth ini menggunakan transistor untuk merubah *level* tegangan yang hanya 3,3 volt menjadi 5 volt sehingga bisa langsung digunakan atau dipasang pada Arduino. HC-05 berkomunikasi dengan Arduino Mega menggunakan komunikasi Serial. Sebelum terhubung dengan Arduino Mega, HC-05 perlu di cek terlebih dahulu berapa nilai *baudrate* (kecepatan pengiriman data) kemudian disesuaikan dengan *baudrate* yang ada pada Arduino Mega karena jenis komunikasi ini adalah *Synchronous*. Jadi kecepatan transmisinya harus sama. HC-05 ini nantinya digunakan sebagai perantara dengan Android yang juga digunakan sebagai *interface* Kontrol dan Monitoring Lampu PJU-TS.





**Gambar 3.12** *Wiring Bluetooth HC-05*

Berikut penjelasan penggunaan masing-masing Pin pada Bluetooth HC-05 dengan Arduino Mega 2560 ditunjukkan pada Tabel 3.5.

**Tabel 3.5** *Wiring Bluetooth HC-05 dengan Arduino Mega 2560*

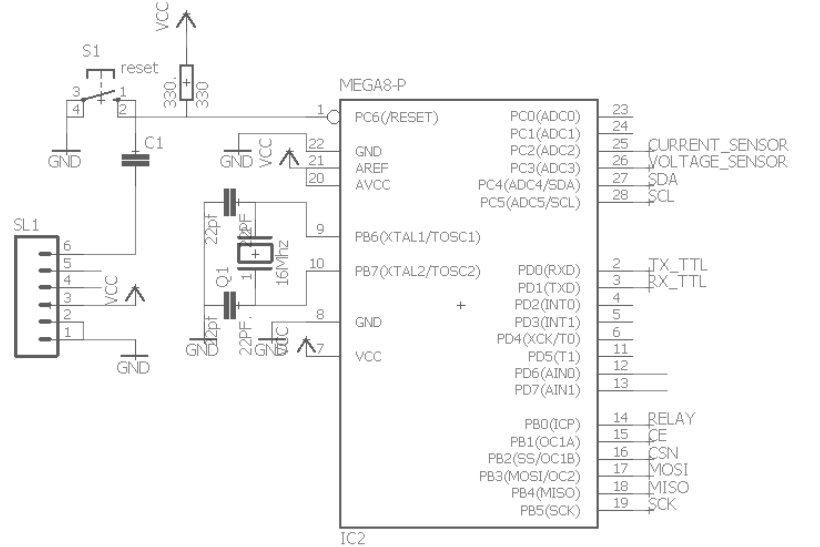
No.	Pin Bluetooth HC-05	Pin Arduino Mega 2560
1.	TX HC-05	RX Serial.3
2.	RX HC-05	TX Serial.3

HC-05 digunakan untuk berkomunikasi dengan Android yang nantinya digunakan sebagai *Interface* untuk melakukan perintah Kontrol dan Monitoring, sehingga dapat di Monitoring dengan mudah oleh Operator. Program pada Android dibuat melalui bantuan *software* MIT App Inventor versi 2.

### 3.1.7 Rangkaian Arduino Nano

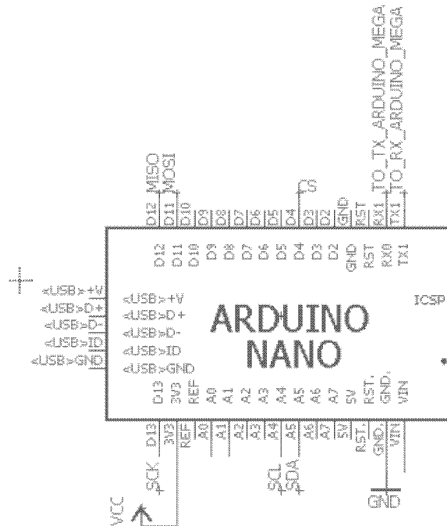
Arduino Nano 328 digunakan sebagai Kontroler kedua dari Master. Fungsi dari Arduino Nano ini digunakan untuk menyimpan data Arus dan Tegangan ke SD Card. Data pembacaan sensor dikirim dari masing-masing sensor dan diterima Arduino Mega yang kemudian dikirim lagi ke Arduino Nano melalui Komunikasi Serial. Sebelumnya data tersebut dibuat ke dalam bentuk paket data kemudian baru dikirim ke Arduino Nano karena nilai data tersebut lebih dari satu yaitu Slave pengirim, Tegangan dari Solar Cell dan Arus Lampu PJU-TS. Setelah data tersebut diterima oleh Arduino nano. Data tersebut kemudian dilakukan parsing untuk memecah kembali sesuai dengan data semula untuk mendapatkan

dua variable yaitu Tegangan Solar Cell dan Arus Lampu PJU-TS. Kemudian data tersebut baru disimpan didalam SD Card. Skematik rangkaian dapat dilihat di Gambar 3.13.



**Gambar 3.13** Rangkaian Arduino Nano

Gambaran dari *Wiring* Arduino Nano dengan perangkat lainnya adalah seperti Gambar 3.14 :



**Gambar 3.14** Fungsi pin Arduino Nano

Komunikasi yang digunakan antara Arduino Mega dengan Arduino Nano menggunakan Komunikasi *wire* yaitu *Universal Serial Asynchronous RX-TX (USART)*, jadi sebelum komunikasi pastikan terlebih dahulu kedua *device* Arduino Mega dan Arduino Nano menggunakan *baudrate* yang sama karena komunikasi perlu sinkronisasi untuk berkomunikasi.

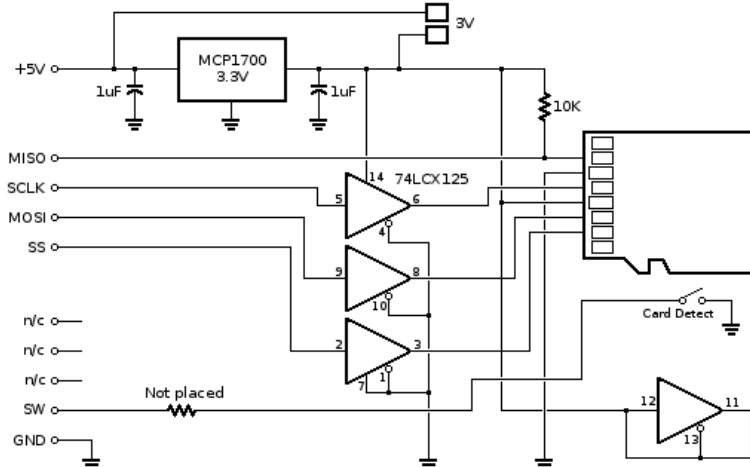
Berikut penjelasan penggunaan masing-masing Pin pada Arduino Nano dengan *device* lain ditunjukkan pada Tabel 3.6.

**Tabel 3.6** Wiring Arduino Nano 328 dengan Arduino Mega 2560

No.	Pin Arduino Nano 328	Keterangan
1.	Serial	Terima data dari Arduino Mega 2560
2.	SPI	Komunikasi dengan SD-Card
3.	I2c	Komunikasi dengan RTC Ds1307

### 3.1.8 Rangkaian Catalex SD Card

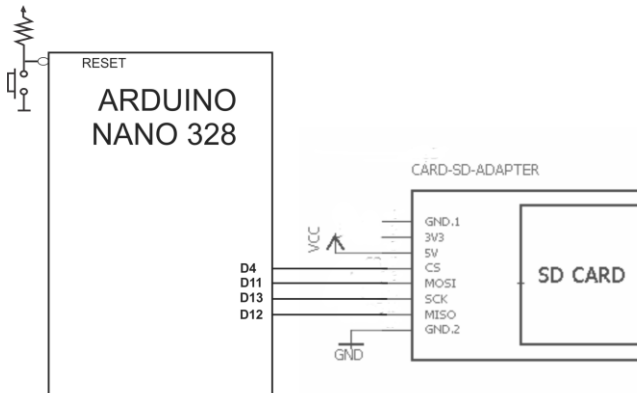
Rangkaian ini merupakan rangkaian Modul Catalex SD, begitu juga dengan SD Card juga menggunakan antarmuka SPI (*Serial Peripheral Interface*) untuk berkomunikasi dengan Arduino nano. Catalex SD Card ini berfungsi untuk menyimpan data tegangan dan Arus yang di terima Arduino Mega dari masing-masing Slave kemudian dikirim kembali oleh Arduino Mega ke Arduino nano, baru kemudian data tersebut disimpan dalam bentuk file di SD Card. File yang disimpan dalam SD card tersebut berektensi text (.txt) atau (.csv). keuntungan dalam pembuatan Datalogger ini adalah dapat mengetahui gambaran komprehensif /statistik kondisi dari masing-masing lampu PJU-TS yang dapat membantu dalam hal Perawatan dan pemeliharaan. Skematik rangkaian dapat dilihat di Gambar 3.15.



**Gambar 3.15** Rangkaian Catalex SD Card

Konfigurasi Perangkat keras Arduino Nano dengan Catalex SD Card adalah seperti Gambar 3.16. pin SPI terdiri dari MISO,MOSI,SCK dan CS. MISO (Master In Slave Out) & MOSI (Master Out Slave In) adalah jalur data untuk komunikasi antara master (programmer / downloader, USBasp) dan Slave (IC mikrokontroller). Sesuai dengan namanya, MISO merupakan jalur yang digunakan download untuk menerima data, sedangkan MOSI adalah jalur downloader mengirim data ke IC mikrokontroller. Kedua jalur ini adalah jalur utama yang digunakan downloader dan mikrokontroller berkomunikasi. Untuk

menghindari kesalahan dalam berkomunikasi, maka dibutuhkan sinkronisasi. Sinkronisasi tersebut dilakukan dengan memanfaatkan jalur SCK (atau ada yang disebut SCLK, Serial CLOCK). Data (MISO dan atau MOSI) akan dianggap valid hanya saat SCK dalam keadaan tinggi. *Wiring* Arduino Nano 328 dengan Catalex SD Card ditunjukkan pada Gambar 3.16.



**Gambar 3.16** *Wiring* Catalex SD Card

Berikut penjelasan penggunaan masing-masing Pin pada Catalex SD Card dengan Arduino Nano ditunjukkan pada Tabel 3.7.

**Tabel 3.7** *Wiring* Catalex SD Card dengan Arduino Nano 328

No.	Pin Catalex SD Card	Pin Arduino Nano 328
1.	CS	Pin Digital No. 4
2.	MOSI	Pin Digital No. 11
3.	SCK	Pin Digital No. 13
4.	MISO	Pin Digital No. 12

### 3.1.9 HMI (*Human Machine Interface*)

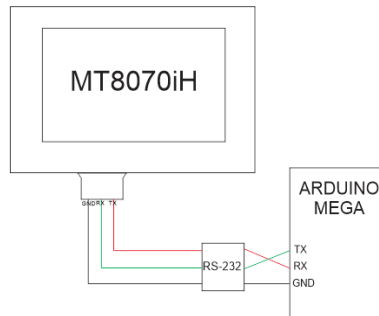
Panel *Touchscreen* WienView adalah alat Kontrol terpusat yang dapat digunakan untuk melakukan perintah Kontrol lampu PJU-TS dan Monitoring data Arus dan tegangan masing-masing lampu PJU secara terpusat. HMI WienView yang dipakai adalah MT6070iH ukuran resolusi pada HMI yaitu 1024 x 600 dengan display 10.1" TF. Power yang digunakan untuk menyalakan HMI adalah 24 VDC. Sedangkan Komunikasi dari HMI ke Arduino menggunakan protokol MODBUS

Serial RS-232. *Human Machine Interface* WienView MT8070iH ditunjukkan pada Gambar 3.17.



**Gambar 3.17** HMI WienView MT8070iH

Protokol Komunikasi Modbus Serial mengatur cara-cara dan format Komunikasi serial (RS-232 atau RS-485) antara Master dengan Slave (Master atau Slave dapat berupa PLC, Mikrokontroler, smart device dll). Wiring diagram WienView MT8070iH dapat dilihat pada Gambar 3.18. *Human Machine Interface* (HMI) tersebut dikoneksikan dengan Arduino Mega Melalui RS-232. Data yang dikirim maupun di Terima Arduino Mega disimpan dalam Holding Register.



**Gambar 3.18** Wiring HMI MT8070iH

Komunikasi antara WienView Weintek MT8070iH dengan Arduino Mega menggunakan Protokol Modbus yaitu dengan memberikan alamat pada Holding Register sesuai dengan perintah kirim atau terima data.

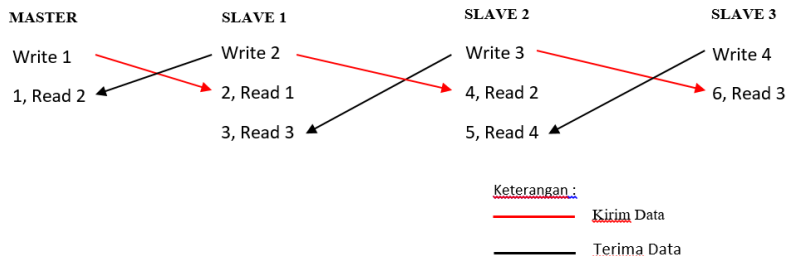
### 3.2 Perancangan Perangkat Lunak

Pada perancangan Perangkat Lunak dibuat dua program untuk menjalankan alat. Program pertama untuk memprogram Arduino dan

program yang lain untuk pemrograman *Interface* pada Android menggunakan MIT App Inventor (*offline*) dan *Human Machine Interface* dengan Easy Builder 8000. Berikut adalah penjelasan dari masing-masing program :

### 3.2.1 Metode Pengiriman Menggunakan *Wireless Sensor Network*

Modul *Transceiver* NRF24L01+ sebuah modul Komunikasi jarak jauh yang memanfaatkan pita gelombang RF 2.4GHz ISM (*Industrial, Scientific and Medical*). Modul ini menggunakan antarmuka SPI untuk berkomunikasi. Tegangan kerja dari modul ini adalah 3.3V DC.



**Gambar 3.19** Metode *Wireless Sensor Network*

Gambar 3.19 merupakan cara/metode komunikasi yang kami buat dalam perancangan Tugas Akhir Sistem Kontrol dan Monitoring lampu PJU-TS menggunakan *Wireless* sensor network terdapat Mikrokontroler yang bertugas sebagai Master dan 3 buah Slave. Perintah kirim data dan terima data dibedakan menjadi 2 saluran data (pipe), disimbolkan dengan warna merah (kirim data) dan hitam (terima data). Apabila Master akan melakukan Perintah Kontrol kepada Slave, maka perintah Kontrol tersebut di bedakan melalui ID number pipe *Address* masing-masing Slave. Setiap perintah Kontrol yang dikirim dari Master akan dilakukan pengecekan setiap Slave, jika perintah tersebut bukan untuk Slave tersebut, maka perintah dari Master akan diteruskan ke Slave selanjutnya. Sebaliknya, pengirim data sensor masing-masing Slave. Data yang dikirimkan setiap Slave akan di terima oleh Slave yang berada sebelumnya. Kemudian data tersebut akan diteruskan hingga akhirnya diterima oleh Master. Dengan begitu dengan menggunakan

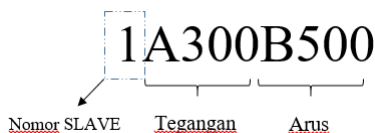
teknik/cara seperti ini memungkinkan mengatasi masalah keterbatasan jarak dan penggunaan radio frekuensi dengan daya yang besar.

Sebelum data dari Slave di kirim ke Master data tersebut di bentuk kedalam suatu paket data, potongan program yang digunakan untuk merubah variabel tegangan solar cell dan arus lampu PJU-TS ditunjukkan pada Gambar 3.20.

```
sprintf(kata,"1A%dB%d",sensorValue,data);  
String katal=kata;
```

**Gambar 3.20** *Sketch* Paket Data Sensor

Data yang dikirim oleh Slave di bentuk kedalam paket Data seperti “1A%dB%d” angka 1 Sebagai Nomor Slave, Huruf A merupakan Nilai Data Tegangan dan Huruf B merupakan Nilai Data Arus. Format pengiriman Paket data yang dikirim oleh Slave dapat dilihat pada Gambar 3.21.



**Gambar 3.21** Paket Data

Setelah data diterima oleh Master, kemudian data tersebut dilakukan parsing. Pertama kali yang dilakukan adalah mengenali Slave Pengirim, pada Gambar 3.21 Slave pengirim ditunjukkan dengan Nomer 1 berarti Paket data tersebut berasal dari Slave 1. Kedua adalah mengidentifikasi Jenis data yang dikirim, Data yang dikirim dari Slave berupa Tegangan dan Arus. Sesuai dengan Gambar 3.21. Nilai ‘A’ Merupakan Tegangan Sedangkan ‘B’ adalah Arus. Setelah dikenali jenis Data tersebut, yang terakhir adalah membaca nilai masing-masing Data. Caranya adalah dengan menggunakan perintah substring, Perintah substring dipakai untuk mengambil string dari nilai awal Sampai nilai akhir.

Pengalamatan yang dilakukan tidak menggunakan IP tapi menggunakan pipe *address*, sebesar 5 byte, ditunjukkan pada Gambar 3.22.



	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0 (RX_ADDR_P0)	0xE7	0xD3	0xF0	0x35	0x77

**Gambar 3.22** Pipe Address NRF24L01

NRF24L01+ memiliki 2 fungsi yaitu fungsi *Transmitter* (Pengirim) dan *Receiver* (Penerima) secara *half duplex*. *Half duplex* Adalah media Komunikasi dua arah. Namun berbeda dengan *full duplex*, *half duplex* berkomunikasi dua arah secara saling bergantian. Jadi saat terjadi Komunikasi antara A dan B. Saat A mengirim informasi (berbicara) maka B akan menerima informasi (mendengarkan). Demikian terjadi proses yang sebaliknya.

Perintah *Transmitter* dan *Receiver* pada Program Arduino dapat dilihat pada Gambar 3.23 dan Gambar 3.24.

Pipe Address  
└───┘  
radio.openWritingPipe(address);

**Gambar 3.23** Sketch *Transmitter* NRF24L01+

Pipe Address  
└───┘  
radio.openReadingPipe(0, address);  
ID Number

**Gambar 3.24** Sketch *Receiver* NRF24L01+

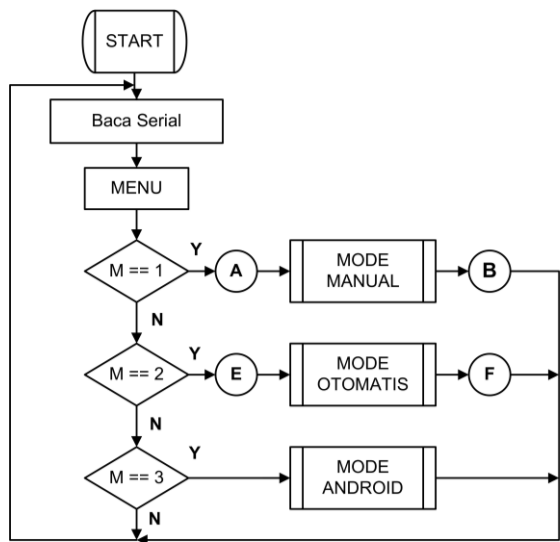
Nilai *address* yang digunakan antara 2 *device* agar dapat berkomunikasi harus sama, Sedangkan ID Number digunakan untuk mengenali *device* mana yang berkomunikasi karena NRF24L01+ dapat berkomunikasi dengan 6 *Receiver* Sekaligus (*MultiReceiver*), dengan kemampuan *MultiReceiver* sehingga NRF24L01+ dapat digunakan untuk perancangan *wireless sensor network*.

### 3.2.2 Algoritma pada Arduino Mega 2560

Program pada Arduino Mega 2560 menggunakan program IDE Arduino. Compiler tersebut digunakan untuk mengkompile bahasa pemrograman Arduino ke bentuk file *hex*.

*Software* yang ditulis menggunakan Arduino dinamakan *sketches*. *Sketches* ini ditulis di editor teks dan disimpan dengan file yang

berekstensi .ino. Editor teks ini mempunyai fasilitas untuk *cut/paste* dan *search/replace*. Area pesan berisi umpan balik ketika menyimpan dan mengunggah file, dan juga menunjukkan jika terjadi *error*. Pada bab ini, dibuat *Flowchart* untuk program Mikrokontroler Arduino Mega 2560 dan Arduino Uno, *Flowchart* program IDE yang terdapat pada Arduino Mega 2560 dapat dilihat pada Gambar 3.25. Arduino Mega digunakan sebagai Master karena mempunyai kapasitas besar dari sisi *flash memory*.



**Gambar 3.25** *Flowchart* Main Program Arduino Mega

Penjelasan *Flowchart* pada Gambar 3.25 jika di tampilkan dalam bentuk program/Sketch ditunjukan pada Gambar 3.26.

```

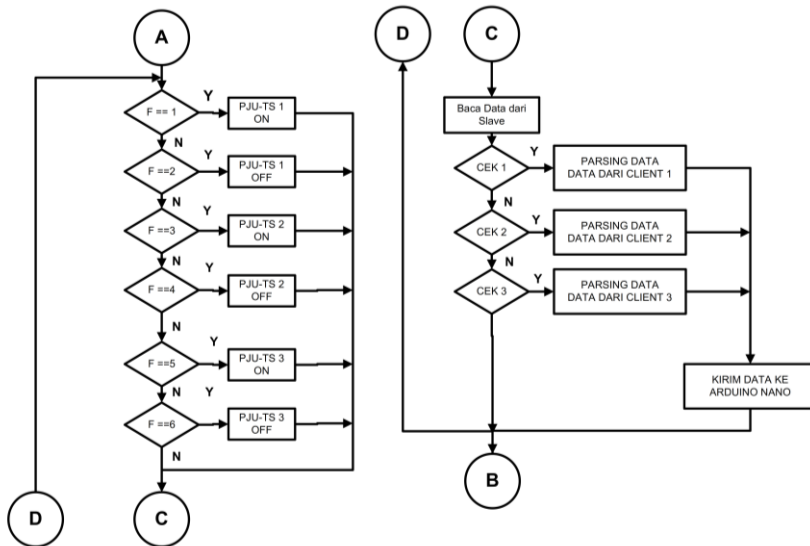
void loop () {
  modbus_update();
  m=holdingRegs[0];
  if (m==1) {
    Manual();
  }
  else if (m==2){
    Otomatis();
  }
  else if (m==3){
    Android();
    delay(1000);
  }
}

```

**Gambar 3.26** Sketch Main Program Arduino

Program Utama yang terdapat pada Arduino Mega yaitu terdiri dari 3 main program yaitu fungsi Manual, Otomatis dan Switch HMI/Android. Perintah “m” digunakan untuk mengendalikan/Kontrol dan Monitoring melalui *Human Machine Interface* (HMI) Wienview Weintek MT8070iH, sedangkan Perintah “f” digunakan untuk mengendalikan/Kontrol dan Monitoring melalui Sistem Operasi Android.

Mode Manual digunakan untuk pengendalian lampu PJU-TS secara manual, operator dapat mengendalikan sesuai dengan keinginan dan kondisi lampu PJU-TS tersebut melalui hasil Monitoring. *Flowchart* Program fungsi Manual ditunjukkan pada Gambar 3.27.



**Gambar 3.27** Flowchart Fungsi Manual Arduino Mega

Pada fungsi ini terdapat 2 sub fungsi yaitu fungsi Kontrol dan Monitoring. Fungsi Kontrol manual dilakukan dengan memilih menu manual pada *Interface* kemudian pilih lampu PJU-TS yang akan di nyalakan/dimatikan. Fungsi ini dideteksi dengan adanya perintah “num” yang menyatakan hubungan Komunikasi kedua *device*, baik antara Master dengan Slave maupun Slave dengan Slave. Fungsi Kontrol digunakan untuk mengendalikan setiap Lampu PJU-TS sedangkan, fungsi Monitoring digunakan untuk memecah data yang dikirimkan oleh setiap Slave kemudian merepresentasikannya kedalam variable Arus Lampu PJU-TS dan Tegangan dari Solar Cell.

```

x=holdingRegs[7];
if(x==3 && mode==1){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=1;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode=0;
}

```

**Gambar 3.28** *Sketch* Kontrol Fungsi Manual Arduino Mega

Program pada fungsi Kontrol “Manual” pertama kali adalah dilakukan memeriksa Protokol Modbus telah siap dan terkoneksi dengan Arduino Mega menggunakan perintah `modbus_update()`. Sedangkan, Komunikasi dengan Android dengan menggunakan perintah `If(Serial3.available()>0)`, karena Bluetooth HC-05 yang digunakan untuk Komunikasi dengan Android Terhubung dengan Serial 3 pada Arduino Mega. Arduino Mega akan membaca perintah data yang dikirim secara serial oleh HMI dan Android dan Arduino mega memberikan perintah kontrol kepada Slave dengan NRF24L01+ melalui perintah `radio.write`. Potongan program Kontrol fungsi manual yang terdapat pada Master dapat dilihat pada Gambar 3.28.

```

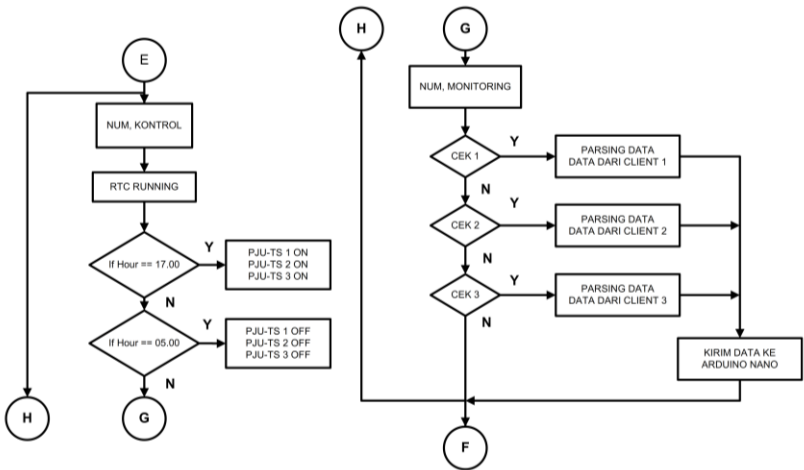
if(num==1){
    radio.read( &kata, sizeof(kata));
    String katal=kata;
    if(katal.charAt(0)=='1'){
        len=katal.length();
        a=katal.indexOf('A');
        b=katal.indexOf('B',a+1);
        sensor=katal.substring(0,a);
        sensor1=katal.substring(a+1,b);
        sensor2=katal.substring(b+1,len);
    }
}

```

**Gambar 3.29** *Sketch* Monitoring Arduino Mega

Program pada fungsi Monitoring “Manual” perintah pertama kali yang dilakukan adalah terima data (`num==1`) artinya Master Terima dari

Slave 1 dengan ID Salve 1. Kemudian dilakukan Pembacaan data oleh NRF24I01+ dengan perintah `radio.read(&kata, sizeof(kata));` ( data yang dikirim NRF24I01+ berformat String). Jika Nilai kata Pertama adalah 1 maka data tersebut berasal dari Slave 1. Jika sesuai dengan Slave pengirimnya kemudian data tersebut dilakukan parsing data, gunananya adalah untuk memecah nilai Tegangan & Arus dari Slave yang dikirimkan masing-masing Slave. Potongan program yang digunakan untuk melakukan parsing data tegangan dan arus ditunjukkan pada Gambar 3.29.



**Gambar 3.30** Flowchart Fungsi Otomatis Arduino Mega

Fungsi Otomatis digunakan untuk mengendalikan Lampu PJU-TS sesuai dengan Jadwal menyala atau mati sesuai dengan waktu yang telah di tentukan. Data waktu tersebut diperoleh dari dari RTC Ds1307, kemudian ketika waktu sesuai dengan jadwal set menyala dan mati Sistem Lampu PJU-TS, NR24L01+ akan memberikan perintah kontrol terhadap Slave. Flowchart Perintah otomatis dapat dilihat pada Gambar 3.30.

```

if(now.hour() == 17 && now.minute() == 0 && now.second() == 0) {
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=1;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
}

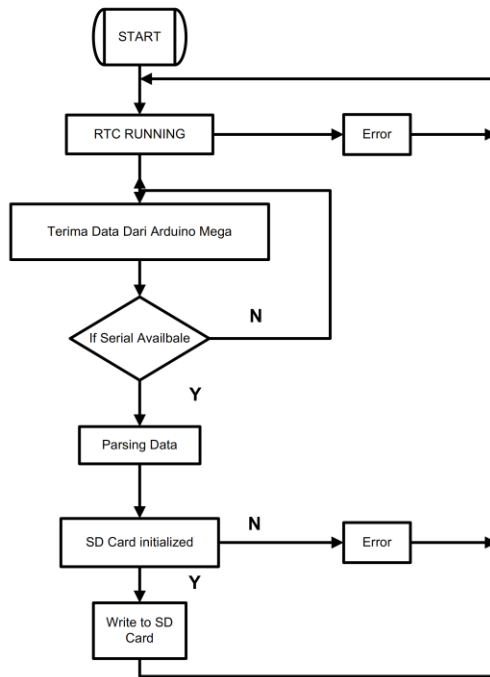
```

**Gambar 3.31** *Sketch* Kontrol Fungsi Otomatis Arduino Mega

Potongan program fungsi kontrol”Otomatis” pada Gambar 3.31 akan melakukan perintah Kontrol setiap masing-masing lampu PJU-TS ketika waktu menunjukan pukul 17.00 Lampu PJU-TS akan Menyala dan Pukul 05.00 Lampu PJU-TS akan mati sesuai dengan waktu *real time clock*.

### 3.2.3 Algoritma pada Arduino Nano

Program pada Arduino Nano juga menggunakan program IDE Arduino, berikut ini adalah Gambar 3.32 yang menunjukan *Flowchart* program pada Arduino Nano.



**Gambar 3.32** Flowchart pada Arduino Nano

Arduino Nano digunakan sebagai datalogger proses pengumpulan dan perekaman Nilai Data tegangan dan Arus masing-masing Slave. Data tersebut dikirim oleh Arduino Mega. Komunikasi yang digunakan untuk mengirim/Komunikasi dari Arduino Mega ke Arduino Nano menggunakan Komunikasi Serial (UART) dengan menggunakan pin TX (transmitter) dan RX (receiver).



```

Wire.begin();
rtc.begin();
if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    rtc.adjust(DateTime(__DATE__, __TIME__))
}
}

```

**Gambar 3.33** *Sketch* Setup RTC

Sebelumnya dilakukan Setup RTC Ds1307, RTC yang digunakan pada Arduino Nano ini bertugas memberika data waktu hasil monitoring sistem Lampu PJU-TS. Gambar 3.33 menunjukan setup RTC Ds1307 pada Arduino Nano. Komunikasi yang digunakan adalah I2c (*Inter Integrated Circuit*). RTC ini digunakan untuk pencatatan data sesuai dengan waktu, jadi kondisi masing-masing lampu dapat dicatat dan disimpan di dalam SD Card. Potongan program setup Catalex SD Card ditunjukkan pada Gambar 3.34.

```

if (!SD.begin(chipSelect)) {

    Serial.println("Card failed, or not present");
    delay(3000);
    return;
}

Serial.println("card initialized.");
delay(2000);

```

**Gambar 3.34** *Sketch* Setup Catalex SD Card

Gambar 3.34 dilakukan untuk setup modul catalex SD Card, Komunikasi yang digunakan adalah SPI (*Serial Peripheral Interface*). Catalex SD Card ini digunakan untuk pencatatan dan perekaman data.

Tahap selanjutnya adalah mengecek port Komunikasi serial Arduino Nano apakah data sudah ada di buffer penerima. Jika data telah diterima oleh Arduino Nano kemudian data tersebut akan dipecah menjadi data hasil Monitoring Lampu PJU-TS yang berupa nilai Tegangan Solar Cell dan Arus Lampu PJU-TS. Jika sudah data tersebut maka akan dilakukan parsing data, parsing data dilakukan dengan membaca panjang string/kata yang diterima kemudian dilakukan

pengenalan melalui paket data tersebut. Susunan paket data yang dikirim oleh Arduino Mega ke bagian Arduino Nano berisi nomor Slave pengirim, Tegangan Solar Cell dan Arus Lampu PJU-TS. kemudian baru disimpan ke dalam SD Card. Potongan program parsing data pada Arduino Nano dapat dilihat pada Gambar 3.35.

```

if(Serial.available()>0){
    res=Serial.readString();
    len=res.length();
    a=res.indexOf('#');
    b=res.indexOf('#',a+1);
    val1=res.substring(0,a);
    val2=res.substring(a+1,b);
    val3=res.substring(b+1,len);
    int x1=val1.indexOf("#");
    int x2=val2.indexOf("#");
    int x3=val3.indexOf("#");
    LogToSD(); |
    }
    delay(1000);
}

```

**Gambar 3.35** *Sketch* Parsing Data

setelah data tersebut diterima kemudian dilakukan parsing dan disimpan ke dalam memory SD Card guna pencatatan dan perekaman data masing-masing Lampu PJU-TS. Perintah Parsing data oleh Arduino Nano bisa dilihat pada Gambar 3.35.

```
File dataFile = SD.open("PJU-TS.txt", FILE_WRITE);
```

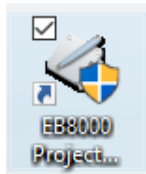
**Gambar 3.6** *Sketch* tulis data di SD Card

Perintah pada Gambar 3.36 digunakan untuk menulis data yang telah di parsing tersebut ke dalam SD Card.

### 3.2.4 Setup Weintek MT8070iH

Perangkat lunak konfigurasi Weintek, Easy Builder, memungkinkan untuk dengan mudah dan cepat membuat proyek dengan objek fungsional termasuk objek numerik, objek Lampu, tombol Combo, Alarm, Recipes, dan banyak lagi. *Human Machne interface* WienView digunakan untuk pengendalian dan Monitoring

lampu PJU-TS. Panel *touchscreen* ini dihubungkan dengan Serial MAX-232 yang terdapat pada Arduino Mega. Pertama kali yang harus dilakukan untuk membuat desain pada Weintek WienView MT8070iH harus melakukan prosedur setup terlebih dahulu, agar *device* yang digunakan bisa berkomunikasi. Langkah pertama adalah buka Easy Builder 8000, *Software* ini digunakan untuk mendesain Panel *Touhscreen* Seri 8000. Pertama kali buka *software* easy builder dengan klik icon seperti Gambar 3.31.



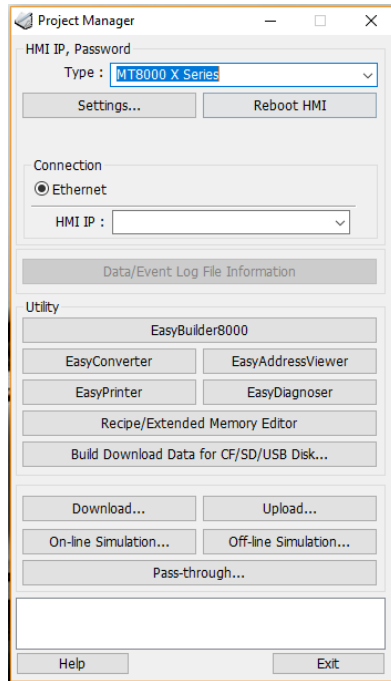
**Gambar 3.31** *Software* Easy Builder 8000

Maka akan muncul jendela baru Project Manager, Lakukan :

Type: MT8000 X Series

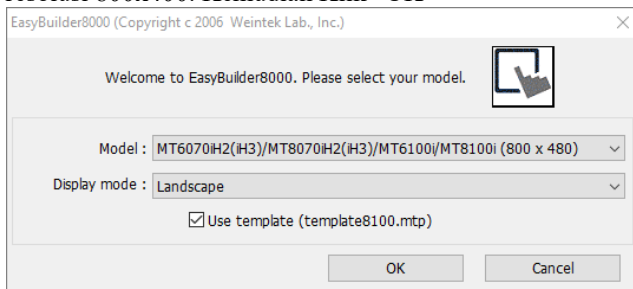
Connection: Ethernet

Kemudian klik “Easy Builder 8000” pada tab utility seperti ditunjukkan pada Gambar 3.32



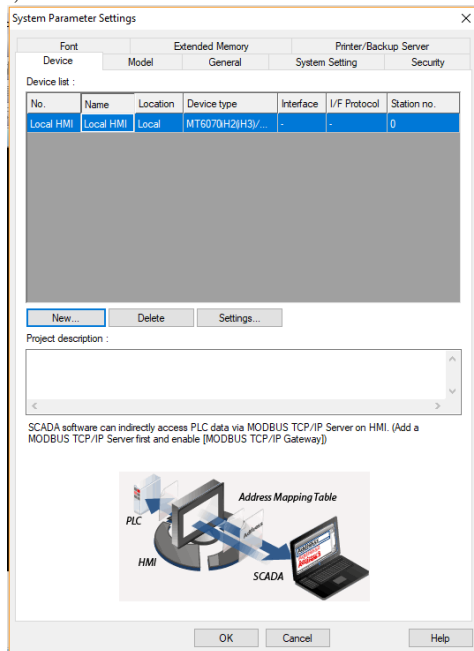
**Gambar 3.32** *Project Manager EB8000*

Muncul Jendela baru seperti ditunjukkan pada Gambar 3.33, lalu pilih Model sesuai dengan *device* yang digunakan yaitu MT8070iH dengan resolusi 800x400. Kemudian Klik “OK”



**Gambar 3.33** *Setup Easy Builder 8000*

Muncul Jendela Sistem Parameter Settings seperti ditunjukkan pada Gambar 3.34, kemudian Klik “New”

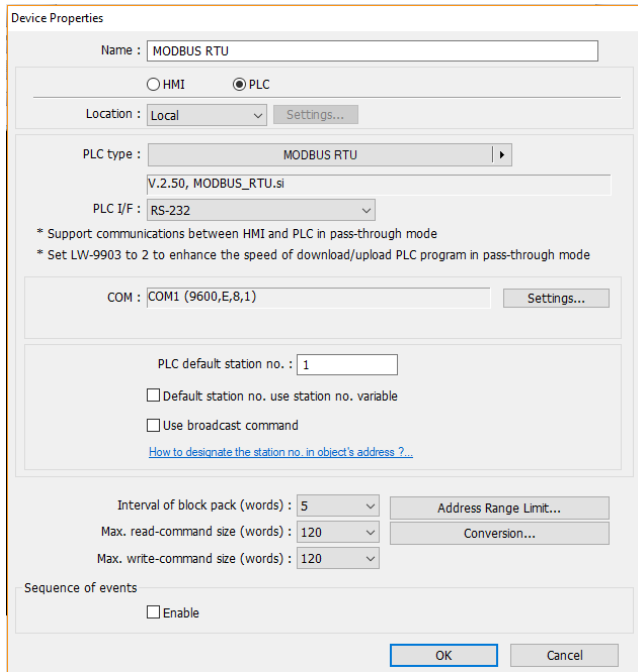


**Gambar 3.34** Sistem Parameter Setting EB8000

Muncul Jendela baru *Device* Propertis seperti ditunjukkan pada Gambar 3.35, Lakukan Setting :

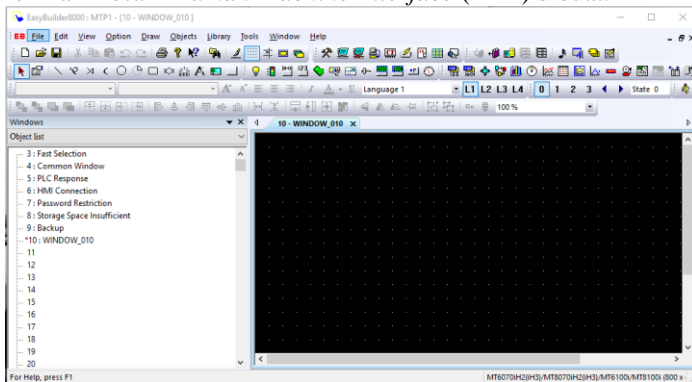
Name : MODBUS RTU HMI  
 Location : Local  
 PLC Type : MODBUS RTU (Protokol)  
 PLC I/F : RS-232 (Komunikasi)  
 COM : COM1 (9600,E,8,1) (baudrate 9600)

Kemudian Klik “OK”



**Gambar 3.35** *Device Properties EB8000*

Kemudian Muncul Window\_10 seperti ditunjukkan pada Gambar 3.36, disinilah Desain *Human Machine Interface* (HMI) dibuat.



**Gambar 3.36** *Tampilan Screen EB8000*

Berikut daftar alamat Holding Register yang digunakan dalam perancangan *Human Machine Interface* pada Tugas Akhir ini ditunjukkan pada Tabel 3.8.

**Tabel 3.8** Address Holding Register Modbus

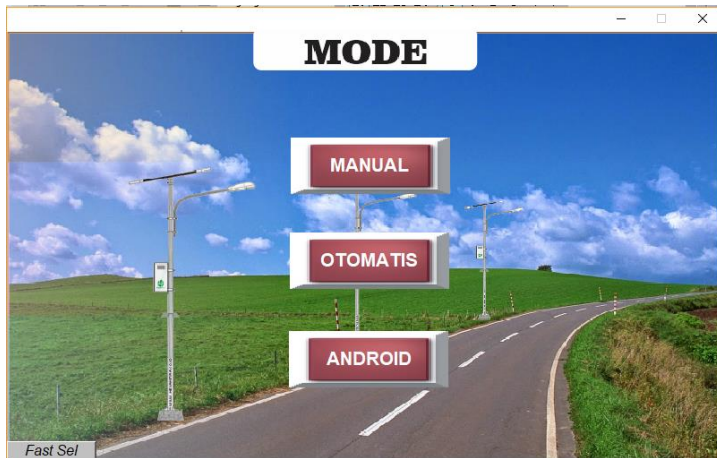
Constant Value	Informasi	Alamat
1	Manual	Holding Register [0]
2	Otomatis	Holding Register [0]
3	Android	Holding Register [0]
3	Kontrol PJU-TS 1 “ON”	Holding Register [7]
4	Kontrol PJU-TS 1 “OFF”	Holding Register [7]
5	Kontrol PJU-TS 2 “ON”	Holding Register [7]
6	Kontrol PJU-TS 2 “OFF”	Holding Register [7]
7	Kontrol PJU-TS 3 “ON”	Holding Register [7]
8	Kontrol PJU-TS 3 “OFF”	Holding Register [7]
-	Data Tegangan PJU-TS 1	Holding Register [1]
-	Data Arus PJU-TS 1	Holding Register [2]
-	Data Tegangan PJU-TS 2	Holding Register [3]
-	Data Arus PJU-TS 2	Holding Register [4]
-	Data Tegangan PJU-TS 3	Holding Register [5]
-	Data Arus PJU-TS 3	Holding Register [6]

Berikut tampilan *Screen Human Machine Interface* MT8070iH :



**Gambar 3.37** Screen 1

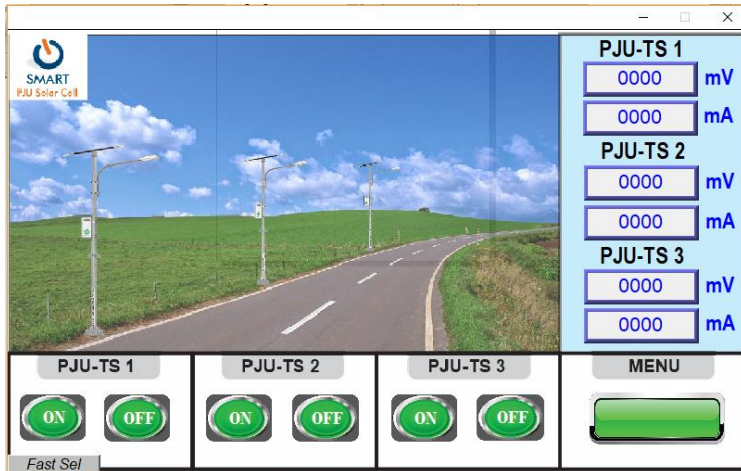
Screen 1 digunakan sebagai tampilan *Human Machine Interface* Sebelum masuk kedalam menu Sistem Lampu PJU-TS.



**Gambar 3.38** Screen 2

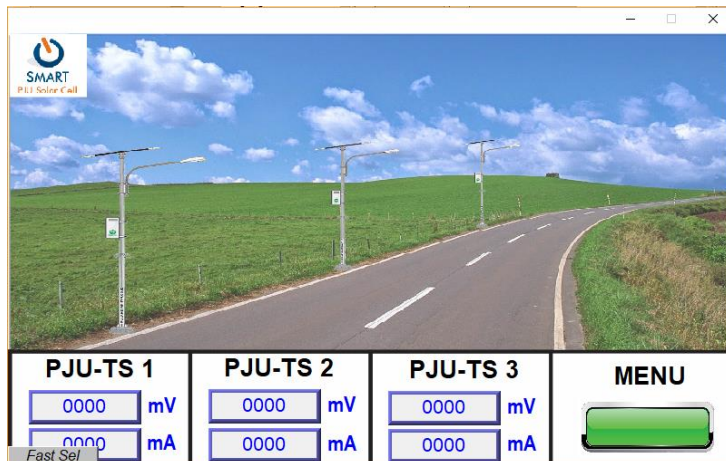
Screen 2 menunjukan Menu yang ada pada *Human Machine Interface*, yaitu terdapat 2 mode pengendalian manual dan otomatis, selain itu juga terdapat Android yang digunakan untuk merubah fungsi pengendalian menggunakan android.





**Gambar 3.39** Screen 3

Screen 3 digunakan sebagai Tampilan Pengendalian Manual dan Monitoring Sistem Lampu PJU-TS.



**Gambar 3.40** Screen 3

Screen 4 digunakan sebagai Tampilan Pengendalian Otomatis dan Monitoring Sistem Lampu PJU-TS.

-----Halaman ini sengaja dikosongkan-----

## **BAB IV**

### **PENGUJIAN DAN ANALISA**

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah tercapai atau belum, maka perlu dilakukannya sebuah pengujian dan analisa terhadap alat yang telah dibuat. Untuk mendapatkan evaluasi terhadap rangkaian dan *software*, sehingga akan dapat dilakukan langkah-langkah positif guna membawa alat ini kearah yang lebih baik. Adapun Tujuan dari pengujian Alat pada Tugas Akhir adalah sebagai berikut :

- Untuk menguji jangkauan jarak Komunikasi NRF24L01+.
- Untuk menguji tingkat keberhasilan Kontrol dan Monitoring Lampu PJU-TS.
- Untuk menguji proses pengumpulan dan pencatatan data pada SD Card.
- Untuk mengetahui penyebab ketidaksempurnaan alat serta menganalisa untuk perbaikan selanjutnya.

Metode pengujian pada Tugas Akhir ini dilakukan meliputi : Pengujian Jarak NRF24L01+, Pengujian Kontrol Lampu PJU-TS, Pengujian Monitoring Lampu PJU-TS, Pengujian *Human Machine Interface* MT8070iH, dan Pengujian Datalogger lampu PJU-TS.

#### **4.1 Pengujian Jarak NRF24L01+**

Pengujian jarak ini dilakukan untuk mengetahui Jarak optimal Komunikasi dengan NRF24L01+. Pada pengujian ini dilakukan sebanyak 100 kali pengiriman Data oleh Slave kemudian Di terima oleh Master. Kemudian Data yang dikirim Oleh Slave dan Master Disesuaikan kemudian diperoleh besarnya Error/Loss Data. Untuk pengujian jarak NRF2401+ dilakukan pengisian program terlebih dahulu menggunakan Arduino IDE. Kemudian Atur Jarak masing-masing antara Slave dengan Master. Catat hasil data kirim dan terima masing-masing.

Peralatan yang dibutuhkan untuk Pengujian adalah sebagai berikut :

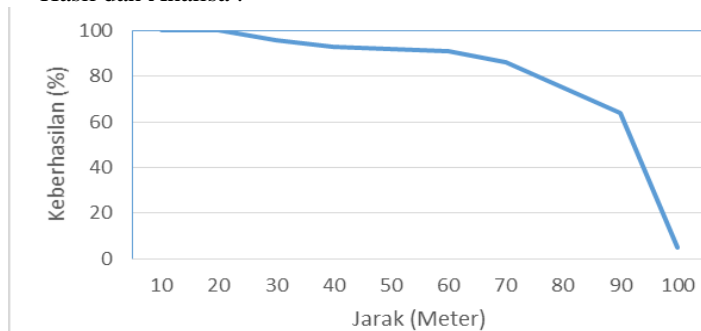
1. PC(Komputer)/Laptop
2. Rangkaian Master dan Slave
3. *Software* Arduino IDE

#### 4. Power Supply



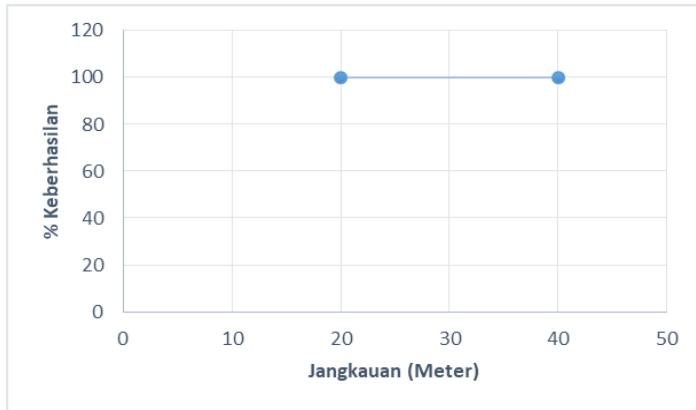
**Gambar 4.1** Pengujian Jarak NRF24L01+

Hasil dan Analisa :



**Grafik 4.1** Pengujian Jarak NRF24L01+ (Tanpa Halangan)

Hasil pengujian jarak NRF24L01+ (Tanpa Halangan) didapatkan hasil seperti Grafik 4.1. Data Tersebut diambil pada 4 Juli 2017. Pengujian dilakukan dengan melakukan kirim data ke Master Sebanyak 20 Data, dengan jarak 10 Meter sampai dengan 100 Meter. Dari pengujian tersebut didapatkan hasil jarak efektif Komunikasi antar Slave dengan Master sejauh 60 Meter dengan presentase keberhasilan 95.33%, Selebihnya ada Data yang dikirim oleh Slave yang mengalami *Loss* Data/Data tersebut tidak sampai ke tujuan. Hal ini bisa terjadi karena Adanya Noise yang mengganggu kualitas dari sinyal dan Delay yang menyebabkan terjadinya antrian penerimaan data pada Master, Delay pengiriman untuk sampai di Master pada Jarak 60 meter mencapai 3 detik.



**Grafik 4.2** Pengujian Jarak NRF24L01+ (Terdapat Halangan)

Hasil pengujian jarak NRF24L01+ (Terdapat Halangan) didapatkan hasil seperti Grafik 4.2. Data tersebut diambil pada 3 Juni 2017. Pengujian dilakukan dengan melakukan kirim data ke Master Sebanyak 20 Data, dengan Halangan berupa Gedung dan Jarak sejauh 20 Meter sampai 40 Meter dengan presentase keberhasilan 100%. Dari pengujian tersebut didapatkan hasil, bahwa pengiriman data dengan jarak sampai 40 meter bekerja dengan baik. Data selengkapnya Hasil Pengujian Jarak NRF24L01+ dapat dilihat pada Lampiran.

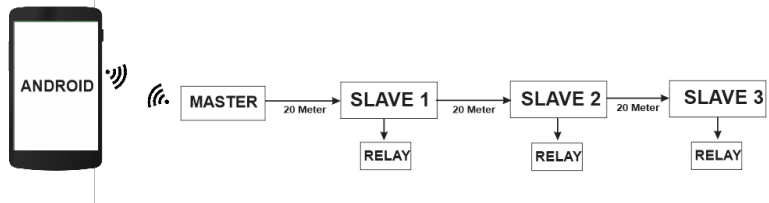
#### **4.2 Pengujian Kontrol Lampu PJU-TS**

Pengujian Kontrol Lampu PJU-TS ini digunakan untuk mengetahui kinerja dari Perintah Kontrol Lampu PJU-TS. Pengujian ini dilakukan dengan cara melakukan Perintah Kontrol dari Master kepada masing-masing Slave dengan rentang jarak 20 Meter. Adapun tahapan yang dilakukan dalam pengujian Kontrol Lampu PJU-TS ini adalah lakukan upload program Slave dan Master. Kemudian atur dan sesuaikan Jarak antar Master dan Slave. Lalu, lakukan perintah kontrol dari Master kepada Slave dengan HMI/Android. Terakhir, lakukan hasil Pengujian Data.

Peralatan yang dibutuhkan untuk Pengujian adalah sebagai berikut :

1. PC(Komputer)
2. Rangkaian Master dan Slave
3. *Software* Arduino IDE

4. *Human Machine Interface/Android*
5. Power Supply



**Gambar 4.2** Pengujian Kontrol Lampu PJU-TS

Hasil dan Analisa :

**Tabel 4.1** Kontrol dari Master ke Slave 1

No.	Perintah Dari MASTER	Perintah Terima SLAVE 1	Jarak antara MASTER dan SLAVE
1.	ON	ON	20 meter
2.	OFF	OFF	20 meter
3.	ON	ON	20 meter
4.	OFF	OFF	20 meter
5.	ON	ON	20 meter
6.	OFF	OFF	20 meter
7.	ON	ON	20 meter
8.	OFF	OFF	20 meter
9.	ON	ON	20 meter
10.	OFF	OFF	20 meter

**Tabel 4.2** Kontrol dari Master ke Slave 2

No.	Perintah Dari MASTER	Perintah Terima SLAVE 2	Jarak antara MASTER dan SLAVE
1.	ON	ON	40 meter
2.	OFF	OFF	40 meter
3.	ON	ON	40 meter
4.	OFF	OFF	40 meter
5.	ON	ON	40 meter
6.	OFF	OFF	40 meter
7.	ON	ON	40 meter
8.	OFF	OFF	40 meter
9.	ON	ON	40 meter
10.	OFF	OFF	40 meter

**Tabel 4.3** Kontrol dari Master ke Slave 3

No.	Perintah Dari MASTER	Perintah Terima SLAVE 3	Jarak antara MASTER dan SLAVE
1.	ON	ON	60 meter
2.	OFF	OFF	60 meter
3.	ON	ON	60 meter
4.	OFF	OFF	60 meter
5.	ON	ON	60 meter
6.	OFF	OFF	60 meter
7.	ON	ON	60 meter
8.	OFF	OFF	60 meter
9.	ON	ON	60 meter
10.	OFF	OFF	60 meter

Hasil pengujian Kontrol Lampu PJU-TS dilakukan dengan perintah Kontrol Lampu PJU-TS sebanyak 20 kali, dengan jarak Master dengan Slave 1 sejauh 20 meter, Master dengan Slave 2 sejauh 40 meter dan Master dengan Slave 3 sejauh 60 meter, diilustrasikan seperti Gambar 4.2. Dari hasil pengujian Kontrol Lampu PJU-TS diperoleh hasil bahwa sistem yang telah di buat bekerja dengan baik, perintah Kontrol dari Master dapat di teruskan oleh setiap Slave sampai ke tujuan. Data Hasil pengujian Kontrol Lampu PJU-TS terhadap setiap Slave ditunjukkan pada Gambar 4.1, 4.2 dan 4.3. Pengambilan Data dilakukan pada tanggal 25 Mei 2017. Data Selengkapnya Hasil Pengujian Kontrol Lampu PJU-TS dapat Dilihat pada Lampiran.

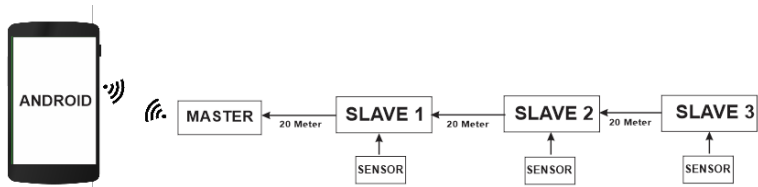
#### **4.3 Pengujian Monitoring Lampu PJU-TS**

Pengujian Monitoring Lampu PJU-TS ini digunakan untuk mengetahui kinerja dan Monitoring data hasil pembacaan masing-masing sensor di bagian Slave. Hasil Monitoring di setiap Slave harus sesuai dengan pengirim maupun penerima. Adapun tahapan yang dilakukan dalam pengujian adalah : pertama lakukan upload program pada masing-masing rangkaian. Sesuaikan jarak seperti ilustrasi pada Gambar 4.3. Terakhir Catat Hasil pembacaan sensor pada Slave kemudian bandingkan data tersebut dengan pembacaan sensor yang ada pada Master Melalui *Human Machine Interface/Android*.

Peralatan yang dibutuhkan untuk Pengujian adalah sebagai berikut :

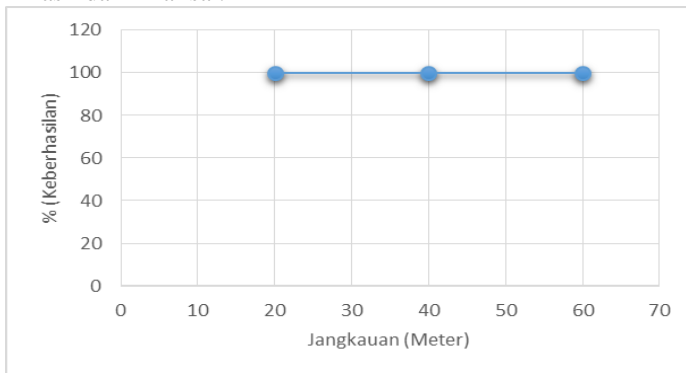
1. PC(Komputer)

2. Rangkaian Master dan Slave
3. *Software* Arduino IDE
4. *Power Supply*
5. *Interface* Android/HMI



**Gambar 4.3** Pengujian Monitoring Lampu PJU-TS

Hasil dan Analisa :



**Grafik 4.3** Grafik Pengujian Monitoring

Hasil Pengujian Monitoring Lampu PJU-TS dengan melakukan pencatatan Kirim Data oleh Slave dan Terima Data pada Master. Kemudian data tersebut dibandingkan, Apakah terdapat Selisih atau ketidaksesuaian antara data yang dikirim dengan data yang diterima. Dari pengujian tersebut ditunjukkan dengan Grafik 4.3. Dari data tersebut Nilai pengirim dan penerima antara Master dengan Slave sesuai. Pengambilan Data dilakukan pada tanggal 25 Mei 2017. Data Selengkapnya Hasil Pengujian Monitoring Lampu PJU-TS dapat Dilihat pada Lampiran.



#### 4.4 Pengujian Mode Otomatis “Kontrol” Lampu PJU-TS

Pengujian Mode Otomatis digunakan untuk menguji fungsi Kontrol otomatis pada Master. Fungsi Kontrol otomatis digunakan untuk pengendalian Lampu PJU-TS sesuai dengan jadwal menyalakan /mematikan Lampu PJU-TS. Adapun tahapan yang dilakukan dalam pengujian adalah: lakukan upload program pada Master dan Slave. Jika sistem telah siap pilih Mode Otomatis yang terdapat pada *Human Machine Interface*/android. Tunggu sesuai dengan pengaturan Waktu pada *Real Time Clock* yang terdapat pada Master. Catat dan sesuaikan dengan waktu kejadian.

Peralatan yang dibutuhkan untuk Pengujian adalah sebagai berikut :

1. PC(Komputer)
2. Rangkaian MASTER dan SLAVE
3. *Software* Arduino IDE
4. *Power Supply*
5. *Interface* Android/HMI

Hasil dan Analisa :

Hasil pengujian Mode Otomatis “Kontrol” Lampu PJU-TS dilakukan dengan pengamatan waktu yang terdapat pada *Real Time Clock* Ds1307 dengan waktu yang sebenarnya. Dari hasil pengujian tersebut ditunjukkan dengan Tabel 4.1. dari table tersebut didapatkan selisih sebesar 15 Detik antara waktu Set RTC Ds1307 dengan waktu sebenarnya. Ketidaktepatan waktu tersebut dikarenakan setting *Real Time Clock* Ds1307 tidak tepat dalam melakukan Set waktu.

**Tabel 4.4** Kontrol Otomatis dari MASTER ke SLAVE 1

No.	Set Waktu	Kondisi		Waktu
1.	12.00.00 WIB.	PJU 1	“ON”	12.00.15 WIB
2.		PJU 2	“ON”	12.00.15 WIB.
3.		PJU 3	“ON”	12.00.15WIB.
4.	01.00.00 WIB.	PJU 1	“OFF”	01.00.15 WIB.
5.		PJU 2	“OFF”	01.00.15 WIB.
6.		PJU 3	“OFF”	01.00.15 WIB.

#### 4.5 Pengujian Human Machine Interface MT8070iH

Pengujian *Human Machine Interface* MT8070iH ini digunakan untuk mengetahui Arduino Mega 2560 dapat berkomunikasi dengan baik dengan HMI, Serta dapat melakukan Perintah Kontrol dan Monitoring terhadap Lampu PJU-TS. Adapun tahapan yang dilakukan dalam pengujian HMI MT8070iH adalah : pertama kali yang dilakukan ada upload program pada Arduino Mega, hubungkan catu daya WienView MT8070iH dengan sumber 24VDC. Kemudian, hubungkan Serial MAX-232 yang terdapat pada Arduino Mega dengan WienView MT8070iH. Lalu Amati Koneksi Antara Arduino dengan HMI.

Peralatan yang dibutuhkan untuk Pengujian adalah sebagai berikut

:

1. PC(Komputer)/Laptop
2. Rangkaian MASTER dan SLAVE
3. *Software* Arduino IDE
4. *Software* Easy Builder 8000
5. Power Supply
6. WienView Weintek MT8070iH

Hasil dan Analisa :



**Gambar 4.4** *Human Machine Interface* MT8070iH

Hasil pengujian WienView MT8070iH dilakukan dengan cara mengkonfigurasi *Human Machine Interface* dengan Arduino Mega 2560 dengan Komunikasi serial menggunakan protokol Modbus. Komunikasi Modbus akan mengatur cara-cara dan format Komunikasi serial dengan 4 jenis penyimpanan data. Jika Perangkat Arduino dengan HMI terkoneksi maka indikator COM pada HMI berkedip, dan perangkat tidak terkoneksi maka akan muncul “PLC no Response” seperti ditunjukkan pada Gambar 4.5.



**Gambar 4.5** Device tidak Terkoneksi dengan HMI

#### 4.6 Pengujian Datalogger Lampu PJU-TS

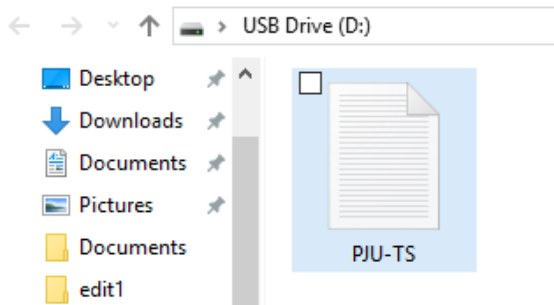
Pengujian Datalogger Lampu PJU-TS ini dilakukan untuk mendapatkan hasil proses otomatis pengumpulan dan perekaman data dari masing-masing Slave. Perekaman tersebut disimpan di dalam SD card Sesuai dengan ID pengirim dan Data dari waktu ke waktu. Adapun tahapan dalam pengujian Datalogger Lampu PJU-TS adalah : Siapkan peralatan yang digunakan, Pastikan data telah diterima oleh Master dengan melihat indikator LED Hijau menyala. Setelah beberapa waktu, matikan *power supply* pada Mikrokontroler. Lepas microSD kemudian masukan pada Card Reader

Peralatan yang dibutuhkan untuk Pengujian adalah sebagai berikut :

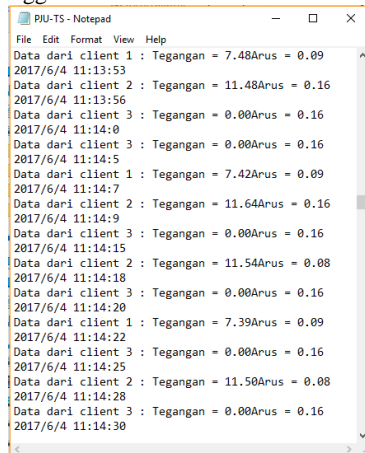
1. PC(Komputer)
2. SD Card penyimpan Datalogger
3. Card Reader

Hasil dan Analisa :

Hasil pengujian datalogger diperoleh pencatatan dan perekaman data hasil Monitoring yang dikirim oleh masing-masing Slave. Data tersebut disimpan dalam format .txt dengan nama file PJU-TS. Ditunjukkan dengan Gambar 4.6. Sedangkan pada Gambar 4.7 menunjukkan Hasil penyimpanan Data yang dikirim Masing-masing Slave sesuai dengan waktu dan tanggal terima.



**Gambar 4.6** Datalogger SD Card



**Gambar 4.7** Hasil Datalogger SD Card

## **BAB V**

### **PENUTUP**

Bab penutup berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

#### **5.1 Kesimpulan**

Berdasarkan pengujian dan analisa perancangan dan implementasi Rancang Bangun Pengendali Utama pada Sistem Kontrol dan Monitoring Lampu PJU-TS menggunakan *Wireless Sensor Network* dengan HMI terpusat. Diperoleh beberapa kesimpulan, antara lain :

1. NRF24L01+ dapat digunakan untuk mengakses *wireless* antara *transmitter* dan *receiver*. Jarak optimal yang digunakan untuk berkomunikasi adalah 50 Meter.
2. Semakin Jauh Jangkauan NRF24L01+ menyebabkan Delay pengiriman yang dapat menyebabkan terjadinya antrian penerimaan data.
3. Metode *Wireless Sensor Network* menggunakan NRF24L01+ dapat berkerja dengan baik, dan mampu menangani ke-tiga Slave dalam pembuatan Tugas Akhir ini.
4. Penempatan antenna NRF24L01+ ditempat di luar box panel Lampu PJU-TS karena, jika di dalam box panel akan menghalangi Komunikasi.
5. Data yang dikirim melalui protokol Modbus di simpan di dalam Holding Register, dengan register alamat 40001 sampai dengan 49999.
6. Pembacaan Data oleh Master dilakukan dengan cara Melakukan Parsing Data String yang di kirim oleh Slave.
7. *Human Machine Interface* WienView MT8070iH dapat memberikan informasi Data Tegangan dan Arus yang dikirim dari masing- masing Slave.
8. Perekaman dan Pencatatan data disimpan di dalam micro SD dengan format PJU-TS.txt

#### **5.2 Saran**

Berikut ini adalah saran-saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini :

1. Sebaiknya dilakukan Setup terlebih dahulu dalam menggunakan NRF24L01+.
2. Untuk pengembangan selanjutnya diharapkan metode pengiriman Data Slave secara *By Request* dan terdapat indikasi saat pengiriman data/terima data.
3. Dalam Mendesain *Human Machine Interface* menggunakan Easy Builder 8000 sebaiknya hindari baground dengan format gambar yang tinggi, karena akan membuat lamanya proses touch.

## DAFTAR PUSTAKA

- [1] Af'idah, Dwi Intan., Adian Fatchur Rochim, Eko Didik Widiyanto., 2014. "Perancangan Jaringan Sensor Nirkabel (Jsn) Untuk Memantau Suhu Dan Kelembaban Menggunakan Nrf24l01+", <http://jtsiskom.undip.ac.id/index.php/jtsiskom/article/view/12600> , 10 Januari 2017
- [2] Arduino, "Arduino Mega 2560", <http://www.arduino.cc/en/Main/arduinoBoardMega2560>, diakses 27 april 2015.
- [3] Anonym. Lampu Penerangan Jalan Umum Tenaga Surya (PJT-TS), <http://repository.usu.ac.id/bitstream/123456789/44711/4/Chapter%20II.pdf> , 25 November 2016
- [4] Anonym. 2015. "Wireless Sensor Network (WSN)", Wikipedia, URL: [https://en.wikipedia.org/wiki/Wireless\\_sensor\\_network](https://en.wikipedia.org/wiki/Wireless_sensor_network), 13 Mei 2015
- [5] Dongre, Manoj M., Swati Rajest Parekar., 2015."Paper An Intelligent System for Monitoring and Controlling of Street Light using GSM Technology", <http://ieeexplore.ieee.org/document/7489455/> , 01 November 2016
- [6] Fajriansyah, Burhan., Muhammad Ichwan, Ratna Susana., 2016. "Evaluasi Karakteristik XBee Pro dan nRF24L01+ sebagai Transceiver Nirkabel", <http://ejurnal.itenas.ac.id/index.php/elkomika/article/view/846> , 11 Januari 2017
- [7] "Kementrian Energi dan Sumber Daya Mineral (2014)". Buku II Perencanaan Sistem PJU Efisien Energi 2014. Jakarta
- [8] Setyawan, Ndaru., "Analisis kinerja sistem lampu jalan pintar dengan tenaga surya berbasis mikrokontroler ". <http://lib.ui.ac.id/abstrakpdfdetail.jsp?id=20386972&lokasi=lokal> , 11 Januari 2017

-----Halaman ini sengaja dikosongkan-----



## LAMPIRAN A

### PENGUJIAN DATA

#### A. Pengujian Jarak (Tanpa Halangan)

1. Pengujian Jarak NRF24L01+ (Tanpa halangan) 10 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 747	Data Diterima = 747	BERHASIL
2.Data ADC = 749	Data Diterima = 749	BERHASIL
3.Data ADC = 759	Data Diterima = 759	BERHASIL
4.Data ADC = 762	Data Diterima = 762	BERHASIL
5.Data ADC = 744	Data Diterima = 744	BERHASIL
6.Data ADC = 748	Data Diterima = 748	BERHASIL
7.Data ADC = 758	Data Diterima = 758	BERHASIL
8.Data ADC = 764	Data Diterima = 764	BERHASIL
9.Data ADC = 754	Data Diterima = 754	BERHASIL
10.Data ADC = 761	Data Diterima = 761	BERHASIL
11.Data ADC = 765	Data Diterima = 765	BERHASIL
12.Data ADC = 757	Data Diterima = 757	BERHASIL
13.Data ADC = 752	Data Diterima = 752	BERHASIL
14.Data ADC = 747	Data Diterima = 747	BERHASIL
15.Data ADC = 742	Data Diterima = 742	BERHASIL
16.Data ADC = 746	Data Diterima = 746	BERHASIL
17.Data ADC = 754	Data Diterima = 754	BERHASIL
18.Data ADC = 762	Data Diterima = 762	BERHASIL
19.Data ADC = 758	Data Diterima = 758	BERHASIL
20.Data ADC = 744	Data Diterima = 744	BERHASIL
21.Data ADC = 748	Data Diterima = 748	BERHASIL
22.Data ADC = 752	Data Diterima = 752	BERHASIL
23.Data ADC = 752	Data Diterima = 752	BERHASIL
24.Data ADC = 749	Data Diterima = 749	BERHASIL
25.Data ADC = 751	Data Diterima = 751	BERHASIL
26.Data ADC = 752	Data Diterima = 752	BERHASIL
27.Data ADC = 753	Data Diterima = 753	BERHASIL
28.Data ADC = 756	Data Diterima = 756	BERHASIL
29.Data ADC = 757	Data Diterima = 757	BERHASIL
30.Data ADC = 760	Data Diterima = 760	BERHASIL
31.Data ADC = 762	Data Diterima = 762	BERHASIL
32.Data ADC = 742	Data Diterima = 742	BERHASIL

Data Dikirim	Data Diterima	STATUS
33.Data ADC = 745	Data Diterima = 745	BERHASIL
34.Data ADC = 750	Data Diterima = 750	BERHASIL
35.Data ADC = 757	Data Diterima = 757	BERHASIL
36.Data ADC = 748	Data Diterima = 748	BERHASIL
37.Data ADC = 755	Data Diterima = 755	BERHASIL
38.Data ADC = 759	Data Diterima = 759	BERHASIL
39.Data ADC = 763	Data Diterima = 763	BERHASIL
40.Data ADC = 763	Data Diterima = 763	BERHASIL
41.Data ADC = 762	Data Diterima = 762	BERHASIL
42.Data ADC = 762	Data Diterima = 762	BERHASIL
43.Data ADC = 759	Data Diterima = 759	BERHASIL
44.Data ADC = 761	Data Diterima = 761	BERHASIL
45.Data ADC = 763	Data Diterima = 763	BERHASIL
46.Data ADC = 749	Data Diterima = 749	BERHASIL
47.Data ADC = 750	Data Diterima = 750	BERHASIL
48.Data ADC = 744	Data Diterima = 744	BERHASIL
49.Data ADC = 761	Data Diterima = 761	BERHASIL
50.Data ADC = 763	Data Diterima = 763	BERHASIL
51.Data ADC = 764	Data Diterima = 764	BERHASIL
52.Data ADC = 753	Data Diterima = 753	BERHASIL
53.Data ADC = 744	Data Diterima = 744	BERHASIL
54.Data ADC = 760	Data Diterima = 760	BERHASIL
55.Data ADC = 761	Data Diterima = 761	BERHASIL
56.Data ADC = 760	Data Diterima = 760	BERHASIL
57.Data ADC = 762	Data Diterima = 762	BERHASIL
58.Data ADC = 756	Data Diterima = 756	BERHASIL
59.Data ADC = 743	Data Diterima = 743	BERHASIL
60.Data ADC = 750	Data Diterima = 750	BERHASIL
61.Data ADC = 750	Data Diterima = 750	BERHASIL
62.Data ADC = 745	Data Diterima = 745	BERHASIL
63.Data ADC = 745	Data Diterima = 745	BERHASIL
64.Data ADC = 766	Data Diterima = 766	BERHASIL
65.Data ADC = 765	Data Diterima = 765	BERHASIL
66.Data ADC = 757	Data Diterima = 757	BERHASIL
67.Data ADC = 755	Data Diterima = 755	BERHASIL
68.Data ADC = 760	Data Diterima = 760	BERHASIL
69.Data ADC = 768	Data Diterima = 768	BERHASIL
70.Data ADC = 764	Data Diterima = 764	BERHASIL

Data Dikirim	Data Diterima	STATUS
71.Data ADC = 754	Data Diterima = 754	BERHASIL
72.Data ADC = 764	Data Diterima = 764	BERHASIL
73.Data ADC = 767	Data Diterima = 767	BERHASIL
74.Data ADC = 759	Data Diterima = 759	BERHASIL
75.Data ADC = 748	Data Diterima = 748	BERHASIL
76.Data ADC = 749	Data Diterima = 749	BERHASIL
77.Data ADC = 765	Data Diterima = 765	BERHASIL
78.Data ADC = 764	Data Diterima = 764	BERHASIL
79.Data ADC = 757	Data Diterima = 757	BERHASIL
80.Data ADC = 749	Data Diterima = 749	BERHASIL
81.Data ADC = 750	Data Diterima = 750	BERHASIL
82.Data ADC = 759	Data Diterima = 759	BERHASIL
83.Data ADC = 763	Data Diterima = 763	BERHASIL
84.Data ADC = 765	Data Diterima = 765	BERHASIL
85.Data ADC = 768	Data Diterima = 768	BERHASIL
86.Data ADC = 765	Data Diterima = 765	BERHASIL
87.Data ADC = 756	Data Diterima = 756	BERHASIL
88.Data ADC = 767	Data Diterima = 767	BERHASIL
89.Data ADC = 746	Data Diterima = 746	BERHASIL
90.Data ADC = 757	Data Diterima = 757	BERHASIL
91.Data ADC = 768	Data Diterima = 768	BERHASIL
92.Data ADC = 763	Data Diterima = 763	BERHASIL
93.Data ADC = 755	Data Diterima = 755	BERHASIL
94.Data ADC = 749	Data Diterima = 749	BERHASIL
95.Data ADC = 766	Data Diterima = 766	BERHASIL
96.Data ADC = 764	Data Diterima = 764	BERHASIL
97.Data ADC = 755	Data Diterima = 755	BERHASIL
98.Data ADC = 746	Data Diterima = 746	BERHASIL
99.Data ADC = 760	Data Diterima = 760	BERHASIL
100.Data ADC = 751	Data Diterima = 751	BERHASIL
Presentase Keberhasilan		100%

2. Pengujian Jarak NRF24L01+ (Tanpa halangan) 20 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 301	Data Diterima = 301	BERHASIL
2.Data ADC = 301	Data Diterima = 301	BERHASIL
3.Data ADC = 300	Data Diterima = 300	BERHASIL

Data Dikirim	Data Diterima	STATUS
4.Data ADC = 307	Data Diterima = 307	BERHASIL
5.Data ADC = 307	Data Diterima = 307	BERHASIL
6.Data ADC = 306	Data Diterima = 306	BERHASIL
7.Data ADC = 305	Data Diterima = 305	BERHASIL
8.Data ADC = 304	Data Diterima = 304	BERHASIL
9.Data ADC = 302	Data Diterima = 302	BERHASIL
10.Data ADC = 308	Data Diterima = 308	BERHASIL
11.Data ADC = 306	Data Diterima = 306	BERHASIL
12.Data ADC = 306	Data Diterima = 306	BERHASIL
13.Data ADC = 303	Data Diterima = 303	BERHASIL
14.Data ADC = 302	Data Diterima = 302	BERHASIL
15.Data ADC = 301	Data Diterima = 301	BERHASIL
16.Data ADC = 301	Data Diterima = 301	BERHASIL
17.Data ADC = 301	Data Diterima = 301	BERHASIL
18.Data ADC = 304	Data Diterima = 304	BERHASIL
19.Data ADC = 307	Data Diterima = 307	BERHASIL
20.Data ADC = 307	Data Diterima = 307	BERHASIL
21.Data ADC = 305	Data Diterima = 305	BERHASIL
22.Data ADC = 305	Data Diterima = 305	BERHASIL
23.Data ADC = 303	Data Diterima = 303	BERHASIL
24.Data ADC = 303	Data Diterima = 303	BERHASIL
25.Data ADC = 304	Data Diterima = 304	BERHASIL
26.Data ADC = 306	Data Diterima = 306	BERHASIL
27.Data ADC = 306	Data Diterima = 306	BERHASIL
28.Data ADC = 307	Data Diterima = 307	BERHASIL
29.Data ADC = 299	Data Diterima = 299	BERHASIL
30.Data ADC = 299	Data Diterima = 299	BERHASIL
31.Data ADC = 306	Data Diterima = 306	BERHASIL
32.Data ADC = 305	Data Diterima = 305	BERHASIL
33.Data ADC = 300	Data Diterima = 300	BERHASIL
34.Data ADC = 301	Data Diterima = 301	BERHASIL
35.Data ADC = 301	Data Diterima = 301	BERHASIL
36.Data ADC = 300	Data Diterima = 300	BERHASIL
37.Data ADC = 300	Data Diterima = 300	BERHASIL
38.Data ADC = 299	Data Diterima = 299	BERHASIL
39.Data ADC = 300	Data Diterima = 300	BERHASIL
40.Data ADC = 308	Data Diterima = 308	BERHASIL
41.Data ADC = 304	Data Diterima = 304	BERHASIL

Data Dikirim	Data Diterima	STATUS
42.Data ADC = 306	Data Diterima = 306	BERHASIL
43.Data ADC = 300	Data Diterima = 300	BERHASIL
44.Data ADC = 301	Data Diterima = 301	BERHASIL
45.Data ADC = 303	Data Diterima = 303	BERHASIL
46.Data ADC = 302	Data Diterima = 302	BERHASIL
47.Data ADC = 300	Data Diterima = 300	BERHASIL
48.Data ADC = 300	Data Diterima = 300	BERHASIL
49.Data ADC = 301	Data Diterima = 301	BERHASIL
50.Data ADC = 300	Data Diterima = 300	BERHASIL
51.Data ADC = 302	Data Diterima = 302	BERHASIL
52.Data ADC = 306	Data Diterima = 306	BERHASIL
53.Data ADC = 306	Data Diterima = 306	BERHASIL
54.Data ADC = 306	Data Diterima = 306	BERHASIL
55.Data ADC = 305	Data Diterima = 305	BERHASIL
56.Data ADC = 304	Data Diterima = 304	BERHASIL
57.Data ADC = 303	Data Diterima = 303	BERHASIL
58.Data ADC = 301	Data Diterima = 301	BERHASIL
59.Data ADC = 300	Data Diterima = 300	BERHASIL
60.Data ADC = 299	Data Diterima = 299	BERHASIL
61.Data ADC = 298	Data Diterima = 298	BERHASIL
62.Data ADC = 302	Data Diterima = 302	BERHASIL
63.Data ADC = 305	Data Diterima = 305	BERHASIL
64.Data ADC = 306	Data Diterima = 306	BERHASIL
65.Data ADC = 306	Data Diterima = 306	BERHASIL
66.Data ADC = 307	Data Diterima = 307	BERHASIL
67.Data ADC = 306	Data Diterima = 306	BERHASIL
68.Data ADC = 306	Data Diterima = 306	BERHASIL
69.Data ADC = 300	Data Diterima = 300	BERHASIL
70.Data ADC = 301	Data Diterima = 301	BERHASIL
71.Data ADC = 302	Data Diterima = 302	BERHASIL
72.Data ADC = 307	Data Diterima = 307	BERHASIL
73.Data ADC = 307	Data Diterima = 307	BERHASIL
74.Data ADC = 307	Data Diterima = 307	BERHASIL
75.Data ADC = 307	Data Diterima = 307	BERHASIL
76.Data ADC = 306	Data Diterima = 306	BERHASIL
77.Data ADC = 307	Data Diterima = 307	BERHASIL
78.Data ADC = 305	Data Diterima = 305	BERHASIL
79.Data ADC = 306	Data Diterima = 306	BERHASIL

Data Dikirim	Data Diterima	STATUS
80.Data ADC = 306	Data Diterima = 306	BERHASIL
81.Data ADC = 304	Data Diterima = 304	BERHASIL
82.Data ADC = 299	Data Diterima = 299	BERHASIL
83.Data ADC = 302	Data Diterima = 302	BERHASIL
84.Data ADC = 299	Data Diterima = 299	BERHASIL
85.Data ADC = 302	Data Diterima = 302	BERHASIL
86.Data ADC = 306	Data Diterima = 306	BERHASIL
87.Data ADC = 306	Data Diterima = 306	BERHASIL
88.Data ADC = 300	Data Diterima = 300	BERHASIL
89.Data ADC = 306	Data Diterima = 306	BERHASIL
90.Data ADC = 300	Data Diterima = 300	BERHASIL
91.Data ADC = 300	Data Diterima = 300	BERHASIL
92.Data ADC = 304	Data Diterima = 304	BERHASIL
93.Data ADC = 305	Data Diterima = 305	BERHASIL
94.Data ADC = 306	Data Diterima = 306	BERHASIL
95.Data ADC = 306	Data Diterima = 306	BERHASIL
96.Data ADC = 306	Data Diterima = 306	BERHASIL
97.Data ADC = 301	Data Diterima = 301	BERHASIL
98.Data ADC = 299	Data Diterima = 299	BERHASIL
99.Data ADC = 302	Data Diterima = 302	BERHASIL
100.Data ADC = 306	Data Diterima = 306	BERHASIL
Presentase Keberhasilan		100%

### 3. Pengujian Jarak NRF24L01+ (Tanpa halangan) 30 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 301	Data Diterima = 301	BERHASIL
2.Data ADC = 308	Data Diterima = 308	BERHASIL
3.Data ADC = 303	Data Diterima = 303	BERHASIL
4.Data ADC = 308	Data Diterima = 308	BERHASIL
5.Data ADC = 307	Data Diterima = 307	BERHASIL
6.Data ADC = 301	Data Diterima = 301	BERHASIL
7.Data ADC = 308	Data Diterima = 308	BERHASIL
8.Data ADC = 305	Data Diterima = 305	BERHASIL
9.Data ADC = 308	Data Diterima = 308	BERHASIL
10.Data ADC = 301	Data Diterima = 301	BERHASIL
11.Data ADC = 301	Data Diterima = 301	BERHASIL
12.Data ADC = 309	Data Diterima = 309	BERHASIL
13.Data ADC = 301	Data Diterima = 301	BERHASIL

Data Dikirim	Data Diterima	STATUS
14.Data ADC = 308	Data Diterima = 308	BERHASIL
15.Data ADC = 309	Data Diterima = 309	BERHASIL
16.Data ADC = 303	Data Diterima = 303	BERHASIL
17.Data ADC = 303	Data Diterima = 306	BERHASIL
18.Data ADC = 306	Data Diterima = 305	BERHASIL
19.Data ADC = 305	Data Diterima = 305	BERHASIL
20.Data ADC = 305	Data Diterima = 305	BERHASIL
21.Data ADC = 305	Data Diterima = 306	BERHASIL
22.Data ADC = 306	-	LOSS
23.Data ADC = 303	Data Diterima = 303	BERHASIL
24.Data ADC = 306	Data Diterima = 306	BERHASIL
25.Data ADC = 301	Data Diterima = 301	BERHASIL
26.Data ADC = 306	Data Diterima = 306	BERHASIL
27.Data ADC = 302	Data Diterima = 302	BERHASIL
28.Data ADC = 304	Data Diterima = 304	BERHASIL
29.Data ADC = 305	Data Diterima = 305	BERHASIL
30.Data ADC = 307	Data Diterima = 307	BERHASIL
31.Data ADC = 301	Data Diterima = 301	BERHASIL
32.Data ADC = 304	Data Diterima = 304	BERHASIL
33.Data ADC = 301	Data Diterima = 301	BERHASIL
34.Data ADC = 302	Data Diterima = 302	BERHASIL
35.Data ADC = 306	Data Diterima = 306	BERHASIL
36.Data ADC = 306	Data Diterima = 306	BERHASIL
37.Data ADC = 300	Data Diterima = 300	BERHASIL
38.Data ADC = 305	Data Diterima = 305	BERHASIL
39.Data ADC = 305	Data Diterima = 305	BERHASIL
40.Data ADC = 300	Data Diterima = 300	BERHASIL
41.Data ADC = 306	Data Diterima = 306	BERHASIL
42.Data ADC = 304	Data Diterima = 304	BERHASIL
43.Data ADC = 303	Data Diterima = 303	BERHASIL
44.Data ADC = 300	Data Diterima = 300	BERHASIL
45.Data ADC = 306	Data Diterima = 306	BERHASIL
46.Data ADC = 300	Data Diterima = 300	BERHASIL
47.Data ADC = 304	Data Diterima = 304	BERHASIL
48.Data ADC = 300	Data Diterima = 300	BERHASIL
49.Data ADC = 300	Data Diterima = 300	BERHASIL
50.Data ADC = 303	Data Diterima = 303	BERHASIL
51.Data ADC = 306	Data Diterima = 306	BERHASIL

Data Dikirim	Data Diterima	STATUS
52.Data ADC = 299	Data Diterima = 299	BERHASIL
53.Data ADC = 306	Data Diterima = 306	BERHASIL
54.Data ADC = 300	Data Diterima = 300	BERHASIL
55.Data ADC = 299	Data Diterima = 299	BERHASIL
56.Data ADC = 306	Data Diterima = 306	BERHASIL
57.Data ADC = 305	Data Diterima = 305	BERHASIL
58.Data ADC = 301	Data Diterima = 301	BERHASIL
59.Data ADC = 306	Data Diterima = 306	BERHASIL
60.Data ADC = 305	Data Diterima = 305	BERHASIL
61.Data ADC = 299	Data Diterima = 299	BERHASIL
62.Data ADC = 305	Data Diterima = 305	BERHASIL
63.Data ADC = 306	Data Diterima = 306	BERHASIL
64.Data ADC = 300	Data Diterima = 300	BERHASIL
65.Data ADC = 300	Data Diterima = 300	BERHASIL
66.Data ADC = 303	Data Diterima = 303	BERHASIL
67.Data ADC = 307	Data Diterima = 307	BERHASIL
68.Data ADC = 300	Data Diterima = 300	BERHASIL
69.Data ADC = 303	Data Diterima = 303	BERHASIL
70.Data ADC = 302	Data Diterima = 302	BERHASIL
71.Data ADC = 300	Data Diterima = 300	BERHASIL
72.Data ADC = 300	Data Diterima = 300	BERHASIL
73.Data ADC = 300	Data Diterima = 300	BERHASIL
74.Data ADC = 303	Data Diterima = 303	BERHASIL
75.Data ADC = 299	Data Diterima = 299	BERHASIL
76.Data ADC = 302	Data Diterima = 302	BERHASIL
77.Data ADC = 305	Data Diterima = 305	BERHASIL
78.Data ADC = 307	Data Diterima = 307	BERHASIL
79.Data ADC = 304	Data Diterima = 304	BERHASIL
80.Data ADC = 308	Data Diterima = 308	BERHASIL
81.Data ADC = 306	Data Diterima = 306	BERHASIL
82.Data ADC = 307	Data Diterima = 307	BERHASIL
83.Data ADC = 307	Data Diterima = 307	BERHASIL
84.Data ADC = 306	Data Diterima = 306	BERHASIL
85.Data ADC = 303	Data Diterima = 303	BERHASIL
86.Data ADC = 304	-	LOSS
87.Data ADC = 301	Data Diterima = 301	BERHASIL
88.Data ADC = 299	Data Diterima = 299	BERHASIL
89.Data ADC = 306	Data Diterima = 306	BERHASIL



Data Dikirim	Data Diterima	STATUS
90.Data ADC = 306	Data Diterima = 306	BERHASIL
91.Data ADC = 299	Data Diterima = 299	BERHASIL
92.Data ADC = 305	Data Diterima = 305	BERHASIL
93.Data ADC = 306	Data Diterima = 306	BERHASIL
94.Data ADC = 300	Data Diterima = 300	BERHASIL
95.Data ADC = 304	-	LOSS
96.Data ADC = 308	Data Diterima = 308	BERHASIL
97.Data ADC = 308	Data Diterima = 308	BERHASIL
98.Data ADC = 304	Data Diterima = 304	BERHASIL
99.Data ADC = 302	Data Diterima = 302	BERHASIL
100.Data ADC = 301	-	LOSS
Presentase Keberhasilan		96%

4. Pengujian Jarak NRF24L01+ (Tanpa halangan) 40 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 974	Data Diterima = 974	BERHASIL
2.Data ADC = 981	Data Diterima = 981	BERHASIL
3.Data ADC = 996	Data Diterima = 996	BERHASIL
4.Data ADC = 981	Data Diterima = 981	BERHASIL
5.Data ADC = 995	Data Diterima = 995	BERHASIL
6.Data ADC = 966	Data Diterima = 966	BERHASIL
7.Data ADC = 995	Data Diterima = 995	BERHASIL
8.Data ADC = 991	Data Diterima = 995	BERHASIL
9.Data ADC = 995	Data Diterima = 995	BERHASIL
10.Data ADC = 995	-	LOSS
11.Data ADC = 977	Data Diterima = 977	BERHASIL
12.Data ADC = 992	Data Diterima = 992	BERHASIL
13.Data ADC = 981	Data Diterima = 981	BERHASIL
14.Data ADC = 970	Data Diterima = 970	BERHASIL
15.Data ADC = 989	Data Diterima = 989	BERHASIL
16.Data ADC = 977	Data Diterima = 977	BERHASIL
17.Data ADC = 971	Data Diterima = 971	BERHASIL
18.Data ADC = 988	Data Diterima = 988	BERHASIL
19.Data ADC = 977	Data Diterima = 977	BERHASIL
20.Data ADC = 965	Data Diterima = 965	BERHASIL
21.Data ADC = 968	Data Diterima = 968	BERHASIL
22.Data ADC = 985	Data Diterima = 985	BERHASIL
23.Data ADC = 989	Data Diterima = 989	BERHASIL

Data Dikirim	Data Diterima	STATUS
24.Data ADC = 990	Data Diterima = 990	BERHASIL
25.Data ADC = 977	Data Diterima = 977	BERHASIL
26.Data ADC = 995	Data Diterima = 995	BERHASIL
27.Data ADC = 966	Data Diterima = 966	BERHASIL
28.Data ADC = 965	Data Diterima = 965	BERHASIL
29.Data ADC = 966	Data Diterima = 966	BERHASIL
30.Data ADC = 991	Data Diterima = 991	BERHASIL
31.Data ADC = 973	Data Diterima = 973	BERHASIL
32.Data ADC = 979	Data Diterima = 979	BERHASIL
33.Data ADC = 966	Data Diterima = 966	BERHASIL
34.Data ADC = 966	Data Diterima = 966	BERHASIL
35.Data ADC = 969	Data Diterima = 969	BERHASIL
36.Data ADC = 992	Data Diterima = 992	BERHASIL
37.Data ADC = 978	Data Diterima = 985	BERHASIL
38.Data ADC = 985		LOSS
39.Data ADC = 975	Data Diterima = 975	BERHASIL
40.Data ADC = 962	Data Diterima = 962	BERHASIL
41.Data ADC = 982	Data Diterima = 982	BERHASIL
42.Data ADC = 986	Data Diterima = 986	BERHASIL
43.Data ADC = 968	Data Diterima = 968	BERHASIL
44.Data ADC = 965	Data Diterima = 965	BERHASIL
45.Data ADC = 981	Data Diterima = 981	BERHASIL
46.Data ADC = 981	-	LOSS
47.Data ADC = 971	-	LOSS
48.Data ADC = 962	Data Diterima = 962	BERHASIL
49.Data ADC = 990	Data Diterima = 990	BERHASIL
50.Data ADC = 990	Data Diterima = 990	BERHASIL
51.Data ADC = 984	Data Diterima = 984	BERHASIL
52.Data ADC = 968	Data Diterima = 968	BERHASIL
53.Data ADC = 978	Data Diterima = 978	BERHASIL
54.Data ADC = 970	Data Diterima = 970	BERHASIL
55.Data ADC = 961	Data Diterima = 961	BERHASIL
56.Data ADC = 977	Data Diterima = 977	BERHASIL
57.Data ADC = 989	Data Diterima = 989	BERHASIL
58.Data ADC = 987	-	LOSS
59.Data ADC = 968	Data Diterima = 968	BERHASIL
60.Data ADC = 988	Data Diterima = 988	BERHASIL
61.Data ADC = 972	Data Diterima = 972	BERHASIL
62.Data ADC = 978	Data Diterima = 978	BERHASIL

Data Dikirim	Data Diterima	STATUS
63.Data ADC = 979	Data Diterima = 979	BERHASIL
64.Data ADC = 966	Data Diterima = 966	BERHASIL
65.Data ADC = 990	Data Diterima = 990	BERHASIL
66.Data ADC = 994	Data Diterima = 994	BERHASIL
67.Data ADC = 991	Data Diterima = 991	BERHASIL
68.Data ADC = 978	-	LOSS
69.Data ADC = 980	-	LOSS
70.Data ADC = 980	Data Diterima = 980	BERHASIL
71.Data ADC = 985	Data Diterima = 985	BERHASIL
72.Data ADC = 986	Data Diterima = 986	BERHASIL
73.Data ADC = 991	Data Diterima = 991	BERHASIL
74.Data ADC = 966	Data Diterima = 966	BERHASIL
75.Data ADC = 968	Data Diterima = 968	BERHASIL
76.Data ADC = 988	Data Diterima = 988	BERHASIL
77.Data ADC = 972	Data Diterima = 972	BERHASIL
78.Data ADC = 971	Data Diterima = 971	BERHASIL
79.Data ADC = 973	Data Diterima = 973	BERHASIL
80.Data ADC = 978	Data Diterima = 978	BERHASIL
81.Data ADC = 978	Data Diterima = 978	BERHASIL
82.Data ADC = 975	Data Diterima = 975	BERHASIL
83.Data ADC = 971	Data Diterima = 971	BERHASIL
84.Data ADC = 965	Data Diterima = 965	BERHASIL
85.Data ADC = 981	Data Diterima = 981	BERHASIL
86.Data ADC = 988	Data Diterima = 988	BERHASIL
87.Data ADC = 961	Data Diterima = 961	BERHASIL
88.Data ADC = 973	Data Diterima = 973	BERHASIL
89.Data ADC = 970	Data Diterima = 970	BERHASIL
90.Data ADC = 987	Data Diterima = 987	BERHASIL
91.Data ADC = 969	Data Diterima = 969	BERHASIL
92.Data ADC = 982	Data Diterima = 982	BERHASIL
93.Data ADC = 958	Data Diterima = 958	BERHASIL
94.Data ADC = 959	Data Diterima = 959	BERHASIL
95.Data ADC = 972	Data Diterima = 972	BERHASIL
96.Data ADC = 987	Data Diterima = 987	BERHASIL
97.Data ADC = 981	Data Diterima = 981	BERHASIL
98.Data ADC = 988	Data Diterima = 988	BERHASIL
99.Data ADC = 960	Data Diterima = 960	BERHASIL
100.Data ADC = 969	Data Diterima = 969	BERHASIL

	Hasil
Presentase Keberhasilan	93%

5. Pengujian Jarak NRF24L01+ (Tanpa halangan) 50 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 989	Data Diterima = 989	BERHASIL
2.Data ADC = 990	Data Diterima = 990	BERHASIL
3.Data ADC = 969	Data Diterima = 969	BERHASIL
4.Data ADC = 969	Data Diterima = 969	BERHASIL
5.Data ADC = 973	Data Diterima = 973	BERHASIL
6.Data ADC = 980	Data Diterima = 980	BERHASIL
7.Data ADC = 983	Data Diterima = 983	BERHASIL
8.Data ADC = 976	Data Diterima = 976	BERHASIL
9.Data ADC = 993	Data Diterima = 993	BERHASIL
10.Data ADC = 992	Data Diterima = 992	BERHASIL
11.Data ADC = 992	Data Diterima = 992	BERHASIL
12.Data ADC = 994	Data Diterima = 994	BERHASIL
13.Data ADC = 994	Data Diterima = 994	BERHASIL
14.Data ADC = 974	Data Diterima = 974	BERHASIL
15.Data ADC = 995	Data Diterima = 995	BERHASIL
16.Data ADC = 993	Data Diterima = 993	BERHASIL
17.Data ADC = 966	Data Diterima = 966	BERHASIL
18.Data ADC = 991	Data Diterima = 991	BERHASIL
19.Data ADC = 993	Data Diterima = 993	BERHASIL
20.Data ADC = 992	Data Diterima = 992	BERHASIL
21.Data ADC = 991	Data Diterima = 991	BERHASIL
22.Data ADC = 966	Data Diterima = 966	BERHASIL
23.Data ADC = 967	-	LOSS
24.Data ADC = 993	Data Diterima = 993	BERHASIL
25.Data ADC = 981	Data Diterima = 981	BERHASIL
26.Data ADC = 974	Data Diterima = 974	BERHASIL
27.Data ADC = 983	Data Diterima = 983	BERHASIL
28.Data ADC = 987	Data Diterima = 987	BERHASIL
29.Data ADC = 968	Data Diterima = 968	BERHASIL
30.Data ADC = 961	Data Diterima = 961	BERHASIL
31.Data ADC = 960	Data Diterima = 960	BERHASIL
32.Data ADC = 984	Data Diterima = 984	BERHASIL
33.Data ADC = 975	Data Diterima = 975	BERHASIL
34.Data ADC = 974	Data Diterima = 992	BERHASIL

Data Dikirim	Data Diterima	STATUS
35.Data ADC = 992	Data Diterima = 989	BERHASIL
36.Data ADC = 989	Data Diterima = 993	BERHASIL
37.Data ADC = 993	Data Diterima = 992	BERHASIL
38.Data ADC = 992	Data Diterima = 992	BERHASIL
39.Data ADC = 992	-	LOSS
40.Data ADC = 993	Data Diterima = 993	BERHASIL
41.Data ADC = 991	Data Diterima = 991	BERHASIL
42.Data ADC = 969	Data Diterima = 969	BERHASIL
43.Data ADC = 973	Data Diterima = 973	BERHASIL
44.Data ADC = 974	Data Diterima = 974	BERHASIL
45.Data ADC = 975	Data Diterima = 975	BERHASIL
46.Data ADC = 974	Data Diterima = 974	BERHASIL
47.Data ADC = 974	Data Diterima = 974	BERHASIL
48.Data ADC = 979	Data Diterima = 979	BERHASIL
49.Data ADC = 975	Data Diterima = 975	BERHASIL
50.Data ADC = 983	Data Diterima = 983	BERHASIL
51.Data ADC = 989	Data Diterima = 989	BERHASIL
52.Data ADC = 958	Data Diterima = 958	BERHASIL
53.Data ADC = 961	Data Diterima = 961	BERHASIL
54.Data ADC = 986	Data Diterima = 986	BERHASIL
55.Data ADC = 980	Data Diterima = 980	BERHASIL
56.Data ADC = 988	Data Diterima = 988	BERHASIL
57.Data ADC = 967	Data Diterima = 967	BERHASIL
58.Data ADC = 984	Data Diterima = 984	BERHASIL
59.Data ADC = 986	Data Diterima = 986	BERHASIL
60.Data ADC = 990	Data Diterima = 990	BERHASIL
61.Data ADC = 991	Data Diterima = 991	BERHASIL
62.Data ADC = 991	Data Diterima = 991	BERHASIL
63.Data ADC = 985	Data Diterima = 985	BERHASIL
64.Data ADC = 973	Data Diterima = 973	BERHASIL
65.Data ADC = 961	Data Diterima = 961	BERHASIL
66.Data ADC = 968	Data Diterima = 968	BERHASIL
67.Data ADC = 981	Data Diterima = 981	BERHASIL
68.Data ADC = 985	Data Diterima = 985	BERHASIL
69.Data ADC = 990	Data Diterima = 990	BERHASIL
70.Data ADC = 991	Data Diterima = 991	BERHASIL
71.Data ADC = 990	Data Diterima = 990	BERHASIL
72.Data ADC = 986	Data Diterima = 986	BERHASIL

Data Dikirim	Data Diterima	STATUS
73.Data ADC = 976	Data Diterima = 976	BERHASIL
74.Data ADC = 968	Data Diterima = 968	BERHASIL
75.Data ADC = 962	Data Diterima = 962	BERHASIL
76.Data ADC = 973	Data Diterima = 973	BERHASIL
77.Data ADC = 989	Data Diterima = 989	BERHASIL
78.Data ADC = 973	Data Diterima = 973	BERHASIL
79.Data ADC = 971	Data Diterima = 994	BERHASIL
80.Data ADC = 994	Data Diterima = 983	BERHASIL
81.Data ADC = 983	-	LOSS
82.Data ADC = 992	Data Diterima = 992	BERHASIL
83.Data ADC = 987	Data Diterima = 990	BERHASIL
84.Data ADC = 990	Data Diterima = 987	BERHASIL
85.Data ADC = 987	-	LOSS
86.Data ADC = 977	Data Diterima = 977	BERHASIL
87.Data ADC = 969	Data Diterima = 969	BERHASIL
88.Data ADC = 994	-	LOSS
89.Data ADC = 975	Data Diterima = 975	BERHASIL
90.Data ADC = 993	Data Diterima = 993	BERHASIL
91.Data ADC = 967	Data Diterima = 967	BERHASIL
92.Data ADC = 983	Data Diterima = 983	BERHASIL
93.Data ADC = 994	Data Diterima = 994	BERHASIL
94.Data ADC = 987	Data Diterima = 987	BERHASIL
95.Data ADC = 977	Data Diterima = 977	BERHASIL
96.Data ADC = 970	Data Diterima = 970	BERHASIL
97.Data ADC = 988	Data Diterima = 988	BERHASIL
98.Data ADC = 992	-	LOSS
99.Data ADC = 964	-	LOSS
100.Data ADC = 989	-	LOSS
Presentase Keberhasilan		92%

6. Pengujian Jarak NRF24L01+ (Tanpa halangan) 60 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 299	Data Diterima = 299	BERHASIL
2.Data ADC = 298	Data Diterima = 298	BERHASIL
3.Data ADC = 305	Data Diterima = 305	BERHASIL
4.Data ADC = 305	Data Diterima = 305	BERHASIL
5.Data ADC = 301	Data Diterima = 301	BERHASIL
6.Data ADC = 297	Data Diterima = 297	BERHASIL

Data Dikirim	Data Diterima	STATUS
7.Data ADC = 298	Data Diterima = 298	BERHASIL
8.Data ADC = 299	Data Diterima = 299	BERHASIL
9.Data ADC = 305	-	LOSS
10.Data ADC = 300	Data Diterima = 300	BERHASIL
11.Data ADC = 300	Data Diterima = 300	BERHASIL
12.Data ADC = 306	Data Diterima = 306	BERHASIL
13.Data ADC = 299	-	LOSS
14.Data ADC = 304	Data Diterima = 304	BERHASIL
15.Data ADC = 299	Data Diterima = 299	BERHASIL
16.Data ADC = 302	Data Diterima = 302	BERHASIL
17.Data ADC = 303	Data Diterima = 303	BERHASIL
18.Data ADC = 305	Data Diterima = 305	BERHASIL
19.Data ADC = 298	Data Diterima = 298	BERHASIL
20.Data ADC = 298	Data Diterima = 298	BERHASIL
21.Data ADC = 298	Data Diterima = 298	BERHASIL
22.Data ADC = 303	Data Diterima = 303	BERHASIL
23.Data ADC = 304	Data Diterima = 304	BERHASIL
24.Data ADC = 306	Data Diterima = 306	BERHASIL
25.Data ADC = 306	Data Diterima = 306	BERHASIL
26.Data ADC = 303	Data Diterima = 303	BERHASIL
27.Data ADC = 302	Data Diterima = 302	BERHASIL
28.Data ADC = 300	Data Diterima = 300	BERHASIL
29.Data ADC = 297	Data Diterima = 297	BERHASIL
30.Data ADC = 297	Data Diterima = 297	BERHASIL
31.Data ADC = 300	Data Diterima = 300	BERHASIL
32.Data ADC = 301	Data Diterima = 301	BERHASIL
33.Data ADC = 301	Data Diterima = 301	BERHASIL
34.Data ADC = 301	Data Diterima = 301	BERHASIL
35.Data ADC = 303	Data Diterima = 303	BERHASIL
36.Data ADC = 301	Data Diterima = 301	BERHASIL
37.Data ADC = 299	Data Diterima = 299	BERHASIL
38.Data ADC = 297	Data Diterima = 297	BERHASIL
39.Data ADC = 298	Data Diterima = 298	BERHASIL
40.Data ADC = 300	Data Diterima = 300	BERHASIL
41.Data ADC = 303	Data Diterima = 303	BERHASIL
42.Data ADC = 302	Data Diterima = 302	BERHASIL
43.Data ADC = 302	Data Diterima = 302	BERHASIL
44.Data ADC = 304	Data Diterima = 304	BERHASIL

Data Dikirim	Data Diterima	STATUS
45.Data ADC = 305	Data Diterima = 305	BERHASIL
46.Data ADC = 305	Data Diterima = 305	BERHASIL
47.Data ADC = 306	Data Diterima = 306	BERHASIL
48.Data ADC = 299	Data Diterima = 299	BERHASIL
49.Data ADC = 298	Data Diterima = 298	BERHASIL
50.Data ADC = 299	Data Diterima = 299	BERHASIL
51.Data ADC = 298	Data Diterima = 298	BERHASIL
52.Data ADC = 298	Data Diterima = 298	BERHASIL
53.Data ADC = 299	Data Diterima = 299	BERHASIL
54.Data ADC = 300	Data Diterima = 300	BERHASIL
55.Data ADC = 298	Data Diterima = 298	BERHASIL
56.Data ADC = 298	Data Diterima = 298	BERHASIL
57.Data ADC = 298	Data Diterima = 298	BERHASIL
58.Data ADC = 300	Data Diterima = 300	BERHASIL
59.Data ADC = 303	Data Diterima = 303	BERHASIL
60.Data ADC = 299	Data Diterima = 299	BERHASIL
61.Data ADC = 303	Data Diterima = 303	BERHASIL
62.Data ADC = 300	Data Diterima = 300	BERHASIL
63.Data ADC = 298	Data Diterima = 298	BERHASIL
64.Data ADC = 305	Data Diterima = 305	BERHASIL
65.Data ADC = 306	Data Diterima = 306	BERHASIL
66.Data ADC = 305	Data Diterima = 305	BERHASIL
67.Data ADC = 301	Data Diterima = 301	BERHASIL
68.Data ADC = 305	-	LOSS
69.Data ADC = 304	-	LOSS
70.Data ADC = 304	Data Diterima = 304	BERHASIL
71.Data ADC = 303	Data Diterima = 303	BERHASIL
72.Data ADC = 305	Data Diterima = 305	BERHASIL
73.Data ADC = 300	-	LOSS
74.Data ADC = 298	Data Diterima = 298	BERHASIL
75.Data ADC = 301	-	LOSS
76.Data ADC = 302	-	LOSS
77.Data ADC = 301	-	LOSS
78.Data ADC = 299	Data Diterima = 299	BERHASIL
79.Data ADC = 301	Data Diterima = 301	BERHASIL
80.Data ADC = 299	Data Diterima = 299	BERHASIL
81.Data ADC = 297	Data Diterima = 297	BERHASIL
82.Data ADC = 299	Data Diterima = 299	BERHASIL



Data Dikirim	Data Diterima	STATUS
83.Data ADC = 304	Data Diterima = 304	BERHASIL
84.Data ADC = 304	Data Diterima = 304	BERHASIL
85.Data ADC = 299	Data Diterima = 299	BERHASIL
86.Data ADC = 302	Data Diterima = 302	BERHASIL
87.Data ADC = 298	Data Diterima = 298	BERHASIL
88.Data ADC = 299	Data Diterima = 299	BERHASIL
89.Data ADC = 304	Data Diterima = 304	BERHASIL
90.Data ADC = 301	Data Diterima = 301	BERHASIL
91.Data ADC = 300	Data Diterima = 300	BERHASIL
92.Data ADC = 298	Data Diterima = 298	BERHASIL
93.Data ADC = 297	Data Diterima = 297	BERHASIL
94.Data ADC = 305	-	LOSS
95.Data ADC = 304	Data Diterima = 304	BERHASIL
96.Data ADC = 303	Data Diterima = 303	BERHASIL
97.Data ADC = 302	Data Diterima = 302	BERHASIL
98.Data ADC = 302	Data Diterima = 302	BERHASIL
99.Data ADC = 304	Data Diterima = 304	BERHASIL
100.Data ADC = 304	Data Diterima = 304	BERHASIL
Presentase Keberhasilan		91%

7. Pengujian Jarak NRF24L01+ (Tanpa halangan) 70 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 301	Data Diterima = 301	BERHASIL
2.Data ADC = 301	Data Diterima = 301	BERHASIL
3.Data ADC = 303	Data Diterima = 303	BERHASIL
4.Data ADC = 306	Data Diterima = 306	BERHASIL
5.Data ADC = 304	Data Diterima = 304	BERHASIL
6.Data ADC = 308	Data Diterima = 308	BERHASIL
7.Data ADC = 303	Data Diterima = 303	BERHASIL
8.Data ADC = 302	-	LOSS
9.Data ADC = 307	Data Diterima = 307	BERHASIL
10.Data ADC = 304	Data Diterima = 304	BERHASIL
11.Data ADC = 300	Data Diterima = 300	BERHASIL
12.Data ADC = 303	Data Diterima = 303	BERHASIL
13.Data ADC = 307	Data Diterima = 307	BERHASIL
14.Data ADC = 306	Data Diterima = 306	BERHASIL
15.Data ADC = 301	Data Diterima = 301	BERHASIL
16.Data ADC = 302	-	LOSS

Data Dikirim	Data Diterima	STATUS
17.Data ADC = 305	Data Diterima = 305	BERHASIL
18.Data ADC = 307	Data Diterima = 307	BERHASIL
19.Data ADC = 302	Data Diterima = 302	BERHASIL
20.Data ADC = 306	Data Diterima = 306	BERHASIL
21.Data ADC = 306	Data Diterima = 306	BERHASIL
22.Data ADC = 301	Data Diterima = 301	BERHASIL
23.Data ADC = 303	Data Diterima = 303	BERHASIL
24.Data ADC = 304	Data Diterima = 304	BERHASIL
25.Data ADC = 306	Data Diterima = 300	BERHASIL
26.Data ADC = 300	Data Diterima = 305	BERHASIL
27.Data ADC = 305	Data Diterima = 299	BERHASIL
28.Data ADC = 299	Data Diterima = 305	BERHASIL
29.Data ADC = 305	-	LOSS
30.Data ADC = 300	-	LOSS
31.Data ADC = 304	Data Diterima = 304	BERHASIL
32.Data ADC = 305	Data Diterima = 305	BERHASIL
33.Data ADC = 305	Data Diterima = 305	BERHASIL
34.Data ADC = 305	Data Diterima = 305	BERHASIL
35.Data ADC = 302	Data Diterima = 302	BERHASIL
36.Data ADC = 306	Data Diterima = 306	BERHASIL
37.Data ADC = 301	Data Diterima = 301	BERHASIL
38.Data ADC = 297	Data Diterima = 297	BERHASIL
39.Data ADC = 303	Data Diterima = 303	BERHASIL
40.Data ADC = 305	Data Diterima = 305	BERHASIL
41.Data ADC = 306	Data Diterima = 306	BERHASIL
42.Data ADC = 298	Data Diterima = 298	BERHASIL
43.Data ADC = 304	-	LOSS
44.Data ADC = 305	-	LOSS
45.Data ADC = 299	Data Diterima = 299	BERHASIL
46.Data ADC = 303	Data Diterima = 303	BERHASIL
47.Data ADC = 300	Data Diterima = 300	BERHASIL
48.Data ADC = 302	Data Diterima = 302	BERHASIL
49.Data ADC = 306	Data Diterima = 306	BERHASIL
50.Data ADC = 305	Data Diterima = 305	BERHASIL
51.Data ADC = 306	Data Diterima = 306	BERHASIL
52.Data ADC = 307	Data Diterima = 307	BERHASIL
53.Data ADC = 305	-	LOSS
54.Data ADC = 299	Data Diterima = 299	BERHASIL

Data Dikirim	Data Dikirim	STATUS
55.Data ADC = 299	Data Diterima = 299	BERHASIL
56.Data ADC = 300	Data Diterima = 300	BERHASIL
57.Data ADC = 306	Data Diterima = 306	BERHASIL
58.Data ADC = 307	Data Diterima = 307	BERHASIL
59.Data ADC = 307	Data Diterima = 307	BERHASIL
60.Data ADC = 304	-	LOSS
61.Data ADC = 305	-	LOSS
62.Data ADC = 306	Data Diterima = 306	BERHASIL
63.Data ADC = 305	Data Diterima = 305	BERHASIL
64.Data ADC = 307	Data Diterima = 307	BERHASIL
65.Data ADC = 304	Data Diterima = 304	BERHASIL
66.Data ADC = 299	Data Diterima = 299	BERHASIL
67.Data ADC = 300	Data Diterima = 300	BERHASIL
68.Data ADC = 306	Data Diterima = 306	BERHASIL
69.Data ADC = 306	Data Diterima = 306	BERHASIL
70.Data ADC = 306	Data Diterima = 305	BERHASIL
71.Data ADC = 305	-	LOSS
72.Data ADC = 301	Data Diterima = 301	BERHASIL
73.Data ADC = 302	Data Diterima = 302	BERHASIL
74.Data ADC = 302	Data Diterima = 302	BERHASIL
75.Data ADC = 301	Data Diterima = 301	BERHASIL
76.Data ADC = 301	Data Diterima = 301	BERHASIL
77.Data ADC = 304	Data Diterima = 304	BERHASIL
78.Data ADC = 304	Data Diterima = 304	BERHASIL
79.Data ADC = 305	Data Diterima = 305	BERHASIL
80.Data ADC = 298	Data Diterima = 298	BERHASIL
81.Data ADC = 305	Data Diterima = 305	BERHASIL
82.Data ADC = 299	Data Diterima = 299	BERHASIL
83.Data ADC = 298	Data Diterima = 298	BERHASIL
84.Data ADC = 298	Data Diterima = 298	BERHASIL
85.Data ADC = 306	Data Diterima = 306	BERHASIL
86.Data ADC = 306	Data Diterima = 306	BERHASIL
87.Data ADC = 306	Data Diterima = 303	BERHASIL
88.Data ADC = 303	-	LOSS
89.Data ADC = 306	Data Diterima = 306	BERHASIL
90.Data ADC = 305	Data Diterima = 304	BERHASIL
91.Data ADC = 304	Data Diterima = 304	BERHASIL
92.Data ADC = 304	-	LOSS

Data Dikirim	Data Dikirim	STATUS
93.Data ADC = 306	-	LOSS
94.Data ADC = 300	Data Diterima = 300	BERHASIL
95.Data ADC = 302	Data Diterima = 302	BERHASIL
96.Data ADC = 302	Data Diterima = 302	BERHASIL
97.Data ADC = 302	Data Diterima = 302	BERHASIL
98.Data ADC = 303	Data Diterima = 303	BERHASIL
99.Data ADC = 305	Data Diterima = 305	BERHASIL
100.Data ADC = 305	-	LOSS
Presentase Keberhasilan		86%

8. Pengujian Jarak NRF24L01+ (Tanpa halangan) 80 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 305	Data Diterima = 305	BERHASIL
2.Data ADC = 298	Data Diterima = 298	BERHASIL
3.Data ADC = 304	Data Diterima = 304	BERHASIL
4.Data ADC = 305	Data Diterima = 305	BERHASIL
5.Data ADC = 298	Data Diterima = 298	BERHASIL
6.Data ADC = 299	Data Diterima = 299	BERHASIL
7.Data ADC = 302	Data Diterima = 302	BERHASIL
8.Data ADC = 306	Data Diterima = 306	BERHASIL
9.Data ADC = 300	Data Diterima = 300	BERHASIL
10.Data ADC = 303	Data Diterima = 303	BERHASIL
11.Data ADC = 303	Data Diterima = 305	BERHASIL
12.Data ADC = 305	Data Diterima = 299	BERHASIL
13.Data ADC = 303	Data Diterima = 302	BERHASIL
14.Data ADC = 299	Data Diterima = 306	BERHASIL
15.Data ADC = 302	Data Diterima = 302	BERHASIL
16.Data ADC = 306	Data Diterima = 304	BERHASIL
17.Data ADC = 302	Data Diterima = 304	BERHASIL
18.Data ADC = 304	Data Diterima = 304	BERHASIL
19.Data ADC = 304	-	LOSS
20.Data ADC = 304	-	LOSS
21.Data ADC = 300	Data Diterima = 300	BERHASIL
22.Data ADC = 305	Data Diterima = 305	BERHASIL
23.Data ADC = 298	-	LOSS
24.Data ADC = 304	Data Diterima = 304	BERHASIL
25.Data ADC = 302	Data Diterima = 302	BERHASIL
26.Data ADC = 306	Data Diterima = 306	BERHASIL

Data Dikirim	Data Diterima	STATUS
27.Data ADC = 303	-	LOSS
28.Data ADC = 305	-	LOSS
29.Data ADC = 297	-	LOSS
30.Data ADC = 304	-	LOSS
31.Data ADC = 300	-	LOSS
32.Data ADC = 301	Data Diterima = 301	BERHASIL
33.Data ADC = 303	Data Diterima = 303	BERHASIL
34.Data ADC = 304	Data Diterima = 304	BERHASIL
35.Data ADC = 305	Data Diterima = 305	BERHASIL
36.Data ADC = 304	Data Diterima = 304	BERHASIL
37.Data ADC = 299	Data Diterima = 299	BERHASIL
38.Data ADC = 299	Data Diterima = 299	BERHASIL
39.Data ADC = 302	Data Diterima = 302	BERHASIL
40.Data ADC = 300	Data Diterima = 300	BERHASIL
41.Data ADC = 303	Data Diterima = 303	BERHASIL
42.Data ADC = 302	Data Diterima = 302	BERHASIL
43.Data ADC = 304	Data Diterima = 304	BERHASIL
44.Data ADC = 306	Data Diterima = 306	BERHASIL
45.Data ADC = 305	Data Diterima = 305	BERHASIL
46.Data ADC = 304	Data Diterima = 304	BERHASIL
47.Data ADC = 304	Data Diterima = 304	BERHASIL
48.Data ADC = 304	Data Diterima = 304	BERHASIL
49.Data ADC = 302	Data Diterima = 302	BERHASIL
50.Data ADC = 301	Data Diterima = 301	BERHASIL
51.Data ADC = 305	Data Diterima = 305	BERHASIL
52.Data ADC = 306	Data Diterima = 306	BERHASIL
53.Data ADC = 301	Data Diterima = 301	BERHASIL
54.Data ADC = 299	Data Diterima = 299	BERHASIL
55.Data ADC = 306	Data Diterima = 306	BERHASIL
56.Data ADC = 300	Data Diterima = 300	BERHASIL
57.Data ADC = 299	-	LOSS
58.Data ADC = 298	-	LOSS
59.Data ADC = 301	-	LOSS
60.Data ADC = 305	-	LOSS
61.Data ADC = 307	Data Diterima = 307	BERHASIL
62.Data ADC = 307	Data Diterima = 307	BERHASIL
63.Data ADC = 306	Data Diterima = 306	BERHASIL
64.Data ADC = 299	Data Diterima = 299	BERHASIL

Data Dikirim	Data Diterima	STATUS
65.Data ADC = 299	Data Diterima = 299	BERHASIL
66.Data ADC = 298	Data Diterima = 302	BERHASIL
67.Data ADC = 299	Data Diterima = 302	BERHASIL
68.Data ADC = 302	Data Diterima = 303	BERHASIL
69.Data ADC = 302	Data Diterima = 306	BERHASIL
70.Data ADC = 303	Data Diterima = 306	BERHASIL
71.Data ADC = 306	Data Diterima = 306	BERHASIL
72.Data ADC = 306	Data Diterima = 306	BERHASIL
73.Data ADC = 306	Data Diterima = 306	BERHASIL
74.Data ADC = 306	Data Diterima = 304	BERHASIL
75.Data ADC = 306	Data Diterima = 304	BERHASIL
76.Data ADC = 301	Data Diterima = 301	BERHASIL
77.Data ADC = 305	-	LOSS
78.Data ADC = 307	-	LOSS
79.Data ADC = 300	-	LOSS
80.Data ADC = 301	-	LOSS
81.Data ADC = 303	-	LOSS
82.Data ADC = 304	Data Diterima = 304	BERHASIL
83.Data ADC = 302	-	LOSS
84.Data ADC = 304	-	LOSS
85.Data ADC = 301	-	LOSS
86.Data ADC = 304	-	LOSS
87.Data ADC = 303	Data Diterima = 303	BERHASIL
88.Data ADC = 300	Data Diterima = 300	BERHASIL
89.Data ADC = 298	Data Diterima = 298	BERHASIL
90.Data ADC = 299	Data Diterima = 299	BERHASIL
91.Data ADC = 300	Data Diterima = 300	BERHASIL
92.Data ADC = 304	Data Diterima = 304	BERHASIL
93.Data ADC = 306	Data Diterima = 306	BERHASIL
94.Data ADC = 305	Data Diterima = 305	BERHASIL
95.Data ADC = 305	Data Diterima = 305	BERHASIL
96.Data ADC = 304	-	LOSS
97.Data ADC = 298	-	LOSS
98.Data ADC = 302	Data Diterima = 302	BERHASIL
99.Data ADC = 305	-	LOSS
100.Data ADC = 298	-	LOSS
Presentase Keberhasilan		75%

9. Pengujian Jarak NRF24L01+ (Tanpa halangan) 90 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 182	Data Diterima = 182	BERHASIL
2.Data ADC = 178	Data Diterima = 178	BERHASIL
3.Data ADC = 181	Data Diterima = 181	BERHASIL
4.Data ADC = 180	Data Diterima = 180	BERHASIL
5.Data ADC = 180	Data Diterima = 180	BERHASIL
6.Data ADC = 179	Data Diterima = 179	BERHASIL
7.Data ADC = 181	Data Diterima = 181	BERHASIL
8.Data ADC = 182	Data Diterima = 182	BERHASIL
9.Data ADC = 177	Data Diterima = 177	BERHASIL
10.Data ADC = 178	Data Diterima = 178	BERHASIL
11.Data ADC = 179	Data Diterima = 179	BERHASIL
12.Data ADC = 180	Data Diterima = 180	BERHASIL
13.Data ADC = 181	Data Diterima = 181	BERHASIL
14.Data ADC = 181	Data Diterima = 181	BERHASIL
15.Data ADC = 181	Data Diterima = 181	BERHASIL
16.Data ADC = 180	Data Diterima = 180	BERHASIL
17.Data ADC = 179	Data Diterima = 179	BERHASIL
18.Data ADC = 178	Data Diterima = 178	BERHASIL
19.Data ADC = 182	Data Diterima = 182	BERHASIL
20.Data ADC = 177	Data Diterima = 177	BERHASIL
21.Data ADC = 182	Data Diterima = 182	BERHASIL
22.Data ADC = 181	Data Diterima = 181	BERHASIL
23.Data ADC = 182	Data Diterima = 182	BERHASIL
24.Data ADC = 182	Data Diterima = 182	BERHASIL
25.Data ADC = 182	Data Diterima = 182	BERHASIL
26.Data ADC = 180	Data Diterima = 180	BERHASIL
27.Data ADC = 180	Data Diterima = 180	BERHASIL
28.Data ADC = 177	-	LOSS
29.Data ADC = 181	Data Diterima = 181	BERHASIL
30.Data ADC = 181	Data Diterima = 181	BERHASIL
31.Data ADC = 179	Data Diterima = 179	BERHASIL
32.Data ADC = 180	Data Diterima = 178	BERHASIL
33.Data ADC = 181	Data Diterima = 181	BERHASIL
34.Data ADC = 177	-	LOSS
35.Data ADC = 181	-	LOSS
36.Data ADC = 179	-	LOSS
37.Data ADC = 178	-	LOSS

Data Dikirim	Data Diterima	STATUS
38.Data ADC = 181	-	LOSS
39.Data ADC = 181	-	LOSS
40.Data ADC = 177	-	LOSS
41.Data ADC = 180	-	LOSS
42.Data ADC = 181	-	LOSS
43.Data ADC = 181	-	LOSS
44.Data ADC = 179	-	LOSS
45.Data ADC = 177	-	LOSS
46.Data ADC = 179	Data Diterima = 179	BERHASIL
47.Data ADC = 181	Data Diterima = 181	BERHASIL
48.Data ADC = 181	Data Diterima = 182	BERHASIL
49.Data ADC = 179	Data Diterima = 181	BERHASIL
50.Data ADC = 179	Data Diterima = 178	BERHASIL
51.Data ADC = 177	Data Diterima = 182	BERHASIL
52.Data ADC = 180	-	LOSS
53.Data ADC = 181	-	LOSS
54.Data ADC = 181	-	LOSS
55.Data ADC = 182	-	LOSS
56.Data ADC = 180	-	LOSS
57.Data ADC = 178	-	LOSS
58.Data ADC = 181	-	LOSS
59.Data ADC = 182	-	LOSS
60.Data ADC = 181	-	LOSS
61.Data ADC = 178	-	LOSS
62.Data ADC = 182	-	LOSS
63.Data ADC = 177	Data Diterima = 177	BERHASIL
64.Data ADC = 181	Data Diterima = 181	BERHASIL
65.Data ADC = 181	-	LOSS
66.Data ADC = 183	Data Diterima = 183	BERHASIL
67.Data ADC = 181	Data Diterima = 181	BERHASIL
68.Data ADC = 181	Data Diterima = 181	BERHASIL
69.Data ADC = 181	Data Diterima = 181	BERHASIL
70.Data ADC = 181	Data Diterima = 181	BERHASIL
71.Data ADC = 177	Data Diterima = 177	BERHASIL
72.Data ADC = 178	Data Diterima = 178	BERHASIL
73.Data ADC = 180	-	LOSS
74.Data ADC = 177	Data Diterima = 177	BERHASIL
75.Data ADC = 181	Data Diterima = 181	BERHASIL



Data Dikirim	Data Diterima	STATUS
76.Data ADC = 179	-	LOSS
77.Data ADC = 177	Data Diterima = 177	BERHASIL
78.Data ADC = 181	Data Diterima = 181	BERHASIL
79.Data ADC = 181	Data Diterima = 181	BERHASIL
80.Data ADC = 178	Data Diterima = 178	BERHASIL
81.Data ADC = 178	Data Diterima = 178	BERHASIL
82.Data ADC = 183	Data Diterima = 183	BERHASIL
83.Data ADC = 181	Data Diterima = 181	BERHASIL
84.Data ADC = 178	Data Diterima = 178	BERHASIL
85.Data ADC = 181	-	LOSS
86.Data ADC = 181	Data Diterima = 181	BERHASIL
87.Data ADC = 179	Data Diterima = 179	BERHASIL
88.Data ADC = 182	-	LOSS
89.Data ADC = 180	-	LOSS
90.Data ADC = 179	-	LOSS
91.Data ADC = 182	-	LOSS
92.Data ADC = 182	-	LOSS
93.Data ADC = 179	-	LOSS
94.Data ADC = 180	-	LOSS
95.Data ADC = 182	-	LOSS
96.Data ADC = 179	-	LOSS
97.Data ADC = 178	Data Diterima = 179	BERHASIL
98.Data ADC = 181	Data Diterima = 181	BERHASIL
99.Data ADC = 181	Data Diterima = 181	BERHASIL
100.Data ADC = 178	Data Diterima = 178	BERHASIL
Presentase Keberhasilan		64%

10. Pengujian Jarak NRF24L01+ (Tanpa halangan) 100 Meter

Data Dikirim	Data Diterima	STATUS
1.Data ADC = 180	Data Diterima = 182	BERHASIL
2.Data ADC = 182	Data Diterima = 182	BERHASIL
3.Data ADC = 182	Data Diterima = 182	BERHASIL
4.Data ADC = 179	Data Diterima = 178	BERHASIL
5.Data ADC = 182	-	LOSS
6.Data ADC = 179	-	LOSS
7.Data ADC = 180	-	LOSS
8.Data ADC = 182	-	LOSS
9.Data ADC = 178	-	LOSS

Data Dikirim	Data Diterima	STATUS
10.Data ADC = 180	-	LOSS
11.Data ADC = 182	-	LOSS
12.Data ADC = 182	-	LOSS
13.Data ADC = 181	-	LOSS
14.Data ADC = 178	-	LOSS
15.Data ADC = 181	-	LOSS
16.Data ADC = 183	-	LOSS
17.Data ADC = 179	-	LOSS
18.Data ADC = 181	Data Diterima = 181	BERHASIL
19.Data ADC = 182	-	LOSS
20.Data ADC = 179	-	LOSS
21.Data ADC = 182	-	LOSS
22.Data ADC = 180	-	LOSS
23.Data ADC = 182	-	LOSS
24.Data ADC = 179	-	LOSS
25.Data ADC = 180	-	LOSS
26.Data ADC = 182	-	LOSS
27.Data ADC = 180	-	LOSS
28.Data ADC = 178	-	LOSS
29.Data ADC = 180	-	LOSS
30.Data ADC = 182	-	LOSS
31.Data ADC = 180	-	LOSS
32.Data ADC = 177	-	LOSS
33.Data ADC = 179	-	LOSS
34.Data ADC = 180	-	LOSS
35.Data ADC = 181	-	LOSS
36.Data ADC = 182	-	LOSS
37.Data ADC = 182	-	LOSS
38.Data ADC = 181	-	LOSS
39.Data ADC = 177	-	LOSS
40.Data ADC = 179	-	LOSS
41.Data ADC = 181	-	LOSS
42.Data ADC = 182	-	LOSS
43.Data ADC = 179	-	LOSS
44.Data ADC = 177	-	LOSS
45.Data ADC = 180	-	LOSS
46.Data ADC = 182	-	LOSS
47.Data ADC = 178	-	LOSS

Data Dikirim	Data Diterima	STATUS
48.Data ADC = 179	-	LOSS
49.Data ADC = 182	-	LOSS
50.Data ADC = 178	-	LOSS
51.Data ADC = 179	-	LOSS
52.Data ADC = 182	-	LOSS
53.Data ADC = 179	-	LOSS
54.Data ADC = 179	-	LOSS
55.Data ADC = 181	-	LOSS
56.Data ADC = 182	-	LOSS
57.Data ADC = 178	-	LOSS
58.Data ADC = 178	-	LOSS
59.Data ADC = 181	-	LOSS
60.Data ADC = 182	-	LOSS
61.Data ADC = 179	-	LOSS
62.Data ADC = 178	-	LOSS
63.Data ADC = 181	-	LOSS
64.Data ADC = 180	-	LOSS
65.Data ADC = 179	-	LOSS
66.Data ADC = 181	-	LOSS
67.Data ADC = 178	-	LOSS
68.Data ADC = 179	-	LOSS
69.Data ADC = 181	-	LOSS
70.Data ADC = 180	-	LOSS
71.Data ADC = 177	-	LOSS
72.Data ADC = 180	-	LOSS
73.Data ADC = 182	-	LOSS
74.Data ADC = 178	-	LOSS
75.Data ADC = 178	-	LOSS
76.Data ADC = 181	-	LOSS
77.Data ADC = 179	-	LOSS
78.Data ADC = 179	-	LOSS
79.Data ADC = 181	-	LOSS
80.Data ADC = 177	-	LOSS
81.Data ADC = 181	-	LOSS
82.Data ADC = 179	-	LOSS
83.Data ADC = 178	-	LOSS
84.Data ADC = 181	-	LOSS
85.Data ADC = 179	-	LOSS

Data Dikirim	Data Diterima	STATUS
86.Data ADC = 178	-	LOSS
87.Data ADC = 181	-	LOSS
88.Data ADC = 177	-	LOSS
89.Data ADC = 181	-	LOSS
90.Data ADC = 178	-	LOSS
91.Data ADC = 181	-	LOSS
92.Data ADC = 179	-	LOSS
93.Data ADC = 180	-	LOSS
94.Data ADC = 178	-	LOSS
95.Data ADC = 181	-	LOSS
96.Data ADC = 178	-	LOSS
97.Data ADC = 181	-	LOSS
98.Data ADC = 179	-	LOSS
99.Data ADC = 179	-	LOSS
100.Data ADC = 181	-	LOSS
Presentase Keberhasilan		5%

## B. Pengujian Jarak (Terdapat Halangan)

### 1. Pengujian Jarak NRF24L01+ (Terdapat halangan) 20 Meter

No.	Kirim Data ADC	Terima Data ADC	Jarak
1.	515	515	20 Meter
2.	307	307	20 Meter
3.	132	132	20 Meter
4.	9	9	20 Meter
5.	311	311	20 Meter
6.	106	106	20 Meter
7.	106	106	20 Meter
8.	106	106	20 Meter
9.	0	0	20 Meter
10.	77	76	20 Meter
11	77	77	20 Meter
12	223	223	20 Meter
13	399	399	20 Meter
14	564	564	20 Meter
15.	957	957	20 Meter
16.	1023	1023	20 Meter
17.	694	694	20 Meter

No.	Kirim Data ADC	Terima Data ADC	Jarak
18.	694	694	20 Meter
19.	455	455	20 Meter
20.	779	779	20 Meter

2. Pengujian Jarak NRF24L01+ (Terdapat halangan) 40 Meter

No.	Kirim Data ADC	Terima Data ADC	Jarak
1.	244	244	40 Meter
2.	435	435	40 Meter
3.	307	307	40 Meter
4.	174	174	40 Meter
5.	0	0	40 Meter
6.	236	237	40 Meter
7.	489	489	40 Meter
8.	501	501	40 Meter
9.	981	981	40 Meter
10.	630	630	40 Meter
11.	468		40 Meter
12.	259	259	40 Meter
13.	259	259	40 Meter
14.	19	19	40 Meter
15.	20	19	40 Meter
16.	324	324	40 Meter
17.	527	527	40 Meter
18.	527	527	40 Meter
19.	909	909	40 Meter
20.	585	585	40 Meter

-----Halaman ini sengaja dikosongkan-----

## **LAMPIRAN B**

### ***LISTING PROGRAM***

**a. Program Arduino Mega “MASTER”**

```
#include <TimerOne.h>
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Wire.h>
#include "RTCLib.h"
#include <SimpleModbusSlave.h>

RF24 radio(9,53);

RTC_DS1307 RTC;

unsigned long data_kirim = 128;
unsigned int data_terima = 128;
unsigned long data = 128;
unsigned int holdingRegs[10];

int m,f,x,a,b,len,stegangan,sarus,tegangan,arus;
const uint64_t pipes[4] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL
    , 0xF0F0F0F0D3LL , 0xF0F0F0F0D4LL };
typedef enum { role_ping_out = 1, role_pong_back } role_e;
const char* role_friendly_name[] = { "invalid", "Ping out", "Pong
    back" };
role_e role = role_pong_back;
byte num;
char buff[16];
int mode=0,mode2=0,mode3=0;
uint8_t kata[16];
String sensor,sensor1,sensor2;
float Tegangan1,Arus1,Tegangan2,Arus2,Tegangan3,Arus3;

void setup (){
    modbus_configure(&Serial1, 9600, SERIAL_8N2, 1, 2, 10,
        holdingRegs);
    Serial.begin(9600);
```

```

Serial2.begin(9600);
Serial3.begin(9600);
radio.begin();
Wire.begin();
RTC.begin();
pinMode(A6,OUTPUT);
pinMode(A7,OUTPUT);
digitalWrite(A6,LOW);
digitalWrite(A7,LOW);
radio.setRetries(15,15);
radio.printDetails();
radio.openWritingPipe(pipes[2]);
radio.openReadingPipe(1,pipes[0]);
radio.stopListening();

if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
}
}

void loop () {
    modbus_update();
    m=holdingRegs[0];
    if (m==1) {
        Manual();
    }

    else if (m==2){
        Otomatis();
    }
    else if (m==3){
        Android();
        delay(1000);
    }
    if (Serial3.available()>0)
    {
        f = Serial3.read();
        if(f=='A'){
            manual();

```



```

    }
    else if(f=='B'){
        otomatis();
    }
    else if(f=='C'){
        HMI();
        delay(1000);
    }
}
}

void Manual(){
    modbus_update();
    radio.startListening();
    if(radio.available(&num)){
        digitalWrite(A6,HIGH);
        digitalWrite(A7,LOW);
        delay(200);
        if(num==1){
            radio.read( &kata, sizeof(kata));
            String kata1=kata;
            if(kata1.charAt(0)=='1'){
                len=kata1.length();
                a=kata1.indexOf('A');
                b=kata1.indexOf('B',a+1);
                sensor=kata1.substring(0,a);
                sensor1=kata1.substring(a+1,b);
                sensor2=kata1.substring(b+1,len);
                stegangan = sensor1.toInt();
                sarus = sensor2.toInt();
                Tegangan1 = (stegangan * (5.0 / 1023.0)*3);
                Arus1      =abs((sarus      *      0.033049784980958577)-
17.061103439792220);
                tegangan = (Tegangan1*1000);
                arus =(Arus1*1000);
                holdingRegs[1] = tegangan;
                holdingRegs[2] = arus;
                Serial2.print("Data dari client 1 : ");
                Serial2.print("#");
            }
        }
    }
}

```

```

Serial2.print("Tegangan = ");
Serial2.print(Tegangan1);
Serial2.print("#");
Serial2.print("Arus = ");
Serial2.println(Arus1);
}

else if(kata1.charAt(0)=='2'){
    len=kata1.length();
    a=kata1.indexOf('A');
    b=kata1.indexOf('B',a+1);
    sensor=kata1.substring(0,a);
    sensor1=kata1.substring(a+1,b);
    sensor2=kata1.substring(b+1,len);
    stegangan = sensor1.toInt();
    sarus = sensor2.toInt();
    Tegangan2 = (stegangan * (5.0 / 1023.0)*3);
    Arus2      =abs((sarus      *      0.075004686035615942)-
38.619812558576584);
    tegangan = (Tegangan2*1000);
    arus =(Arus2*1000);
    holdingRegs[3] = tegangan;
    holdingRegs[4] = arus;
    Serial2.print("Data dari client 2 : ");
    Serial2.print("#");
    Serial2.print("Tegangan = ");
    Serial2.print(Tegangan2);
    Serial2.print("#");
    Serial2.print("Arus = ");
    Serial2.println(Arus2);
}

else if(kata1.charAt(0)=='3'){
    len=kata1.length();
    a=kata1.indexOf('A');
    b=kata1.indexOf('B',a+1);
    sensor=kata1.substring(0,a);
    sensor1=kata1.substring(a+1,b);
    sensor2=kata1.substring(b+1,len);

```

```

    stegangan = sensor1.toInt();
    sarus = sensor2.toInt();
    Tegangan3 = (stegangan * (5.0 / 1023.0)*3);
    Arus3      =abs((sarus      *      0.080372183395599942)-
41.389228295583294);
    tegangan = (Tegangan3*1000);
    arus =(Arus3*1000);
    holdingRegs[5] = tegangan;
    holdingRegs[6] = arus;
    Serial2.print("Data dari client 3 : ");
    Serial2.print("#");
    Serial2.print("Tegangan = ");
    Serial2.print(Tegangan3);
    Serial2.print("#");
    Serial2.print("Arus = ");
    Serial2.println(Arus3);
}
Serial3.print(Tegangan1);
Serial3.print("|");
Serial3.print(Arus1);
Serial3.print("|");
Serial3.print(Tegangan2);
Serial3.print("|");
Serial3.print(Arus2);
Serial3.print("|");
Serial3.print(Tegangan3);
Serial3.print("|");
Serial3.print(Arus3);
Serial3.println();
delay(1000);
}
}
x=holdingRegs[7];
if(x==3 && mode==1){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=1;
    radio.write( &data_kirim, sizeof(unsigned long));

```

```

    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode=0;
}
else if(x==4 && mode==0){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=0;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode=1;
}
else if(x==5 && mode2==0){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=2;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode2=1;
}
else if(x==6 && mode2==1){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=3;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);

```

```

    digitalWrite(A7,LOW);
    mode2=0;
}
else if(x==7 && mode3==0){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=4;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode3=1;
}
else if(x==8 && mode3==1){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=5;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode3=0;
}
if (m==2){
    Otomatis();
}
else if (m==3) {
    Android();
    delay(1000);
}
}

void manual(){
    while(true){
        modbus_update();
    }
}

```

```

        radio.startListening();
    if(radio.available(&num)){
        digitalWrite(A6,HIGH);
        digitalWrite(A7,LOW);
        delay(200);
        if(num==1){
            radio.read( &kata, sizeof(kata));
            String kata1=kata;
            if(kata1.charAt(0)=='1'){
                len=kata1.length();
                a=kata1.indexOf('A');
                b=kata1.indexOf('B',a+1);
                sensor=kata1.substring(0,a);
                sensor1=kata1.substring(a+1,b);
                sensor2=kata1.substring(b+1,len);
                stegangan = sensor1.toInt();
                sarus = sensor2.toInt();
                Tegangan1 = (stegangan * (5.0 / 1023.0)*3);
                Arus1      =abs((sarus      *      0.033049784980958577)-
17.061103439792220);
                tegangan = (Tegangan1*1000);
                arus =(Arus1*1000);
                holdingRegs[1] = tegangan;
                holdingRegs[2] = arus;
                Serial2.print("Data dari client 1 : ");
                Serial2.print("#");
                Serial2.print("Tegangan = ");
                Serial2.print(Tegangan1);
                Serial2.print("#");
                Serial2.print("Arus = ");
                Serial2.println(Arus1);
            }

            else if(kata1.charAt(0)=='2'){
                len=kata1.length();
                a=kata1.indexOf('A');
                b=kata1.indexOf('B',a+1);
                sensor=kata1.substring(0,a);
                sensor1=kata1.substring(a+1,b);

```

```

        sensor2=kata1.substring(b+1,len);
        stegangan = sensor1.toInt();
        sarus = sensor2.toInt();
        Tegangan2 = (stegangan * (5.0 / 1023.0)*3);
        Arus2      =abs((sarus      *      0.075004686035615942)-
38.619812558576584);
        tegangan = (Tegangan2*1000);
        arus =(Arus2*1000);
        holdingRegs[3] = tegangan;
        holdingRegs[4] = arus;
        Serial2.print("Data dari client 2 : ");
        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan2);
        Serial2.print("#");
        Serial2.print("Arus = ");
        Serial2.println(Arus2);
    }

    else if(kata1.charAt(0)=='3'){
        len=kata1.length();
        a=kata1.indexOf('A');
        b=kata1.indexOf('B',a+1);
        sensor=kata1.substring(0,a);
        sensor1=kata1.substring(a+1,b);
        sensor2=kata1.substring(b+1,len);
        stegangan = sensor1.toInt();
        sarus = sensor2.toInt();
        Tegangan3 = (stegangan * (5.0 / 1023.0)*3);
        Arus3      =abs((sarus      *      0.080372183395599942)-
41.389228295583294);
        tegangan = (Tegangan3*1000);
        arus =(Arus3*1000);
        holdingRegs[5] = tegangan;
        holdingRegs[6] = arus;
        Serial2.print("Data dari client 3 : ");
        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan3);

```

```

    Serial2.print("#");
    Serial2.print("Arus = ");
    Serial2.println(Arus3);
}

Serial3.print(Tegangan1);
Serial3.print("|");
Serial3.print(Arus1);
Serial3.print("|");
Serial3.print(Tegangan2);
Serial3.print("|");
Serial3.print(Arus2);
Serial3.print("|");
Serial3.print(Tegangan3);
Serial3.print("|");
Serial3.print(Arus3);
Serial3.println();
delay(1000);
}
}
if (Serial3.available()>0)
{
f = Serial3.read();
if(f=='1' && mode==1){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=1;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
    mode=0;
}

else if(f=='2' && mode==0){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);

```



```

radio.stopListening();
data_kirim=0;
radio.write( &data_kirim, sizeof(unsigned long));
radio.startListening();
delay(200);
mode=1;
digitalWrite(A6,LOW);
digitalWrite(A7,LOW);
}
else if(f=='3' && mode2==0){
digitalWrite(A6,LOW);
digitalWrite(A7,HIGH);
radio.stopListening();
data_kirim=2;
radio.write( &data_kirim, sizeof(unsigned long));
radio.startListening();
delay(200);
mode2=1;
digitalWrite(A6,LOW);
digitalWrite(A7,LOW);
}

else if(f=='4' && mode2==1){
digitalWrite(A6,LOW);
digitalWrite(A7,HIGH);
radio.stopListening();
data_kirim=3;
radio.write( &data_kirim, sizeof(unsigned long));
radio.startListening();
delay(200);
mode2=0;
digitalWrite(A6,LOW);
digitalWrite(A7,LOW);
}

else if(f=='5' && mode3==0){
digitalWrite(A6,LOW);
digitalWrite(A7,HIGH);
radio.stopListening();

```

```

    data_kirim=4;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    mode3=1;
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
}

else if(f=='6' && mode3==1){
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=5;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    mode3=0;
    digitalWrite(A6,LOW);
    digitalWrite(A7,LOW);
}

else if(f=='B'){
    otomatis();
}
else if(f=='C'){
    HMI();
    delay(1000);
}
}
}

void Otomatis(){
    modbus_update();
    DateTime now = RTC.now();
    Serial.print(now.day(), DEC);
    Serial.print('-');
    Serial.print(now.month(), DEC);

```

```

Serial.print('-');
Serial.print(now.year(), DEC);
Serial.print(' ');
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
delay(1000);
radio.startListening();
if(radio.available(&num)){
digitalWrite(A6,HIGH);
digitalWrite(A7,LOW);
delay(200);
if(num==1){
radio.read( &kata, sizeof(kata));
String kata1=kata;
if(kata1.charAt(0)=='1'){
len=kata1.length();
a=kata1.indexOf('A');
b=kata1.indexOf('B',a+1);
sensor=kata1.substring(0,a);
sensor1=kata1.substring(a+1,b);
sensor2=kata1.substring(b+1,len);
stegangan = sensor1.toInt();
sarus = sensor2.toInt();
Tegangan1 = (stegangan * (5.0 / 1023.0)*3);
Arus1      =abs((sarus      *      0.033502065342286586)-
17.121758204676453);
tegangan = (Tegangan1*1000);
arus =(Arus1*1000);
holdingRegs[1] = tegangan;
holdingRegs[2] = arus;
Serial2.print("Data dari client 1 : ");
Serial2.print("#");
Serial2.print("Tegangan = ");
Serial2.print(Tegangan1);
Serial2.print("#");

```

```

        Serial2.print("Arus = ");
        Serial2.println(Arus1);
    }
    else if(kata1.charAt(0)=='2'){
        len=kata1.length();
        a=kata1.indexOf('A');
        b=kata1.indexOf('B',a+1);
        sensor=kata1.substring(0,a);
        sensor1=kata1.substring(a+1,b);
        sensor2=kata1.substring(b+1,len);
        stegangan = sensor1.toInt();
        sarus = sensor2.toInt();
        Tegangan2 = (stegangan * (5.0 / 1023.0)*3);
        Arus2      =abs((sarus      *      0.075004686035615942)-
38.619812558576584);
        tegangan = (Tegangan2*1000);
        arus =(Arus2*1000);
        holdingRegs[3] = tegangan;
        holdingRegs[4] = arus;
        Serial2.print("Data dari client 2 : ");
        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan2);
        Serial2.print("#");
        Serial2.print("Arus = ");
        Serial2.println(Arus2);
    }
    else if(kata1.charAt(0)=='3'){
        len=kata1.length();
        a=kata1.indexOf('A');
        b=kata1.indexOf('B',a+1);
        sensor=kata1.substring(0,a);
        sensor1=kata1.substring(a+1,b);
        sensor2=kata1.substring(b+1,len);
        stegangan = sensor1.toInt();
        sarus = sensor2.toInt();
        Tegangan3 = (stegangan * (5.0 / 1023.0)*3);
        Arus3      =abs((sarus      *      0.080372183395599942)-
41.389228295583294);

```

```

        tegangan = (Tegangan3*1000);
        arus =(Arus3*1000);
        holdingRegs[5] = tegangan;
        holdingRegs[6] = arus;
        Serial2.print("Data dari client 3 : ");
        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan3);
        Serial2.print("#");
        Serial2.print("Arus = ");
        Serial2.println(Arus3);
    }
    Serial3.print(Tegangan1);
    Serial3.print("|");
    Serial3.print(Arus1);
    Serial3.print("|");
    Serial3.print(Tegangan2);
    Serial3.print("|");
    Serial3.print(Arus2);
    Serial3.print("|");
    Serial3.print(Tegangan3);
    Serial3.print("|");
    Serial3.print(Arus3);
    Serial3.println();
    delay(1000);
}
}
if(now.hour() == 8 && now.minute() == 45 && now.second()
== 0) {
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=1;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    radio.stopListening();
    data_kirim=2;
    radio.write( &data_kirim, sizeof(unsigned long));

```

```

        radio.startListening();
        delay(200);
        radio.stopListening();
        data_kirim=4;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        digitalWrite(A6,LOW);
        digitalWrite(A7,LOW);
    }
    else if(now.hour() == 8 && now.minute() == 50 &&
now.second() == 0){
        digitalWrite(A6,LOW);
        digitalWrite(A7,HIGH);
        radio.stopListening();
        data_kirim=0;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        radio.stopListening();
        data_kirim=3;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        radio.stopListening();
        data_kirim=5;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        digitalWrite(A6,LOW);
        digitalWrite(A7,LOW);
    }
    if (m==1) {
        Manual();
    }
    else if (m==3){
        Android();
        delay(1000);
    }
}

```

```

}

void otomatis(){
while(true){
DateTime now = RTC.now();
    Serial.print(now.day(), DEC);
    Serial.print('-');
    Serial.print(now.month(), DEC);
    Serial.print('-');
    Serial.print(now.year(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();
    delay(1000);
    radio.startListening();
if(radio.available(&num)){
    digitalWrite(A6,HIGH);
    digitalWrite(A7,LOW);
    delay(200);
    if(num==1){
        radio.read( &kata, sizeof(kata));
        String kata1=kata;
        if(kata1.charAt(0)=='1'){
            len=kata1.length();
            a=kata1.indexOf('A');
            b=kata1.indexOf('B',a+1);
            sensor=kata1.substring(0,a);
            sensor1=kata1.substring(a+1,b);
            sensor2=kata1.substring(b+1,len);
            stegangan = sensor1.toInt();
            sarus = sensor2.toInt();
            Tegangan1 = (stegangan * (5.0 / 1023.0)*3);
            Arus1      =abs((sarus      *      0.033049784980958577)-
17.061103439792220);
            Serial2.print("Data dari client 1 : ");

```

```

        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan1);
        Serial2.print("#");
        Serial2.print("Arus = ");
        Serial2.println(Arus1);
    }

    else if(kata1.charAt(0)=='2'){
        len=kata1.length();
        a=kata1.indexOf('A');
        b=kata1.indexOf('B',a+1);
        sensor=kata1.substring(0,a);
        sensor1=kata1.substring(a+1,b);
        sensor2=kata1.substring(b+1,len);
        stegangan = sensor1.toInt();
        sarus = sensor2.toInt();
        Tegangan2 = (stegangan * (5.0 / 1023.0)*3);
        Arus2      =abs((sarus      *      0.075004686035615942)-
38.619812558576584);
        Serial2.print("Data dari client 2 : ");
        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan2);
        Serial2.print("#");
        Serial2.print("Arus = ");
        Serial2.println(Arus2);
    }

    else if(kata1.charAt(0)=='3'){
        len=kata1.length();
        a=kata1.indexOf('A');
        b=kata1.indexOf('B',a+1);
        sensor=kata1.substring(0,a);
        sensor1=kata1.substring(a+1,b);
        sensor2=kata1.substring(b+1,len);
        stegangan = sensor1.toInt();
        sarus = sensor2.toInt();
        Tegangan3 = (stegangan * (5.0 / 1023.0)*3);

```



```

        Arus3      =abs((sarus      *      0.080372183395599942)-
41.389228295583294);
        Serial2.print("Data dari client 3 : ");
        Serial2.print("#");
        Serial2.print("Tegangan = ");
        Serial2.print(Tegangan3);
        Serial2.print("#");
        Serial2.print("Arus = ");
        Serial2.println(Arus3);
    }
    Serial3.print(Tegangan1);
    Serial3.print("|");
    Serial3.print(Arus1);
    Serial3.print("|");
    Serial3.print(Tegangan2);
    Serial3.print("|");
    Serial3.print(Arus2);
    Serial3.print("|");
    Serial3.print(Tegangan3);
    Serial3.print("|");
    Serial3.print(Arus3);
    Serial3.println();
    delay(1000);
}
}
if(now.hour() == 17 && now.minute() == 30 && now.second()
== 0) {
    digitalWrite(A6,LOW);
    digitalWrite(A7,HIGH);
    radio.stopListening();
    data_kirim=1;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
    radio.stopListening();
    data_kirim=2;
    radio.write( &data_kirim, sizeof(unsigned long));
    radio.startListening();
    delay(200);
}

```

```

        radio.stopListening();
        data_kirim=4;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        digitalWrite(A6,LOW);
        digitalWrite(A7,LOW);
    }
    else if(now.hour() == 4 && now.minute() == 30 &&
now.second() == 0){
        digitalWrite(A6,LOW);
        digitalWrite(A7,HIGH);
        radio.stopListening();
        data_kirim=0;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        radio.stopListening();
        data_kirim=3;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        radio.stopListening();
        data_kirim=5;
        radio.write( &data_kirim, sizeof(unsigned long));
        radio.startListening();
        delay(200);
        digitalWrite(A6,LOW);
        digitalWrite(A7,LOW);
    }
    if (Serial3.available(>0)
    {
        f = Serial3.read();
        if(f=='A'){
            manual();
        }
        else if(f=='C'){
            HMI();
            delay(1000);
        }
    }

```

```

    }
  }
}
}
void Android();{
  if (Serial3.available()>0)
  {
    f = Serial3.read();
    if(f=='A'){
      manual();
    }
    else if(f=='B'){
      otomatis();
    }
  }
}
void HMI();
modbus_update();
m=holdingRegs[0];
if (m==1) {
  Manual();
}

else if (m==2){
  Otomatis();
}
}

```

**b. Program Arduino Nano “Datalogger”**

```

#include <LiquidCrystal.h>
#include <Wire.h>
#include <RTClib.h>
#include <SD.h>

```

```

RTC_DS1307 rtc;

```

```

const int chipSelect = 4;
String res;
int a,b,len;

```

```

String val1, val2, val3;
void setup(){
  Serial.begin(9600);
  Serial.print("Initializing SD card...");
  pinMode(10, OUTPUT);
  digitalWrite(10,HIGH);
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    delay(3000);
    return;
  }

  Serial.println("card initialized.");
  delay(2000);
  Wire.begin();
  rtc.begin();
  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    rtc.adjust(DateTime(__DATE__, __TIME__));
  }
}

void loop() {
  DateTime now = rtc.now();
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  if(Serial.available()>0){
    res=Serial.readString();
    len=res.length();
    a=res.indexOf('#');

```

```

b=res.indexOf('#',a+1);
val1=res.substring(0,a);
val2=res.substring(a+1,b);
val3=res.substring(b+1,len);
int x1=val1.indexOf("#");
int x2=val2.indexOf("#");
int x3=val3.indexOf("#");
Serial.print(val1);
Serial.print(val2);
Serial.println(val3);
LogToSD();
}
delay(1000);
}

```

```

void LogToSD(){
DateTime now = rtc.now();
File dataFile = SD.open("logger.txt", FILE_WRITE);
if (dataFile) {
  dataFile.print(val1);
  dataFile.print(val2);
  dataFile.print(val3);
  dataFile.print(now.year(), DEC);
  dataFile.print('/');
  dataFile.print(now.month(), DEC);
  dataFile.print('/');
  dataFile.print(now.day(), DEC);
  dataFile.print(" ");
  dataFile.print(now.hour(), DEC);
  dataFile.print(':');
  dataFile.print(now.minute(), DEC);
  dataFile.print(':');
  dataFile.print(now.second(), DEC);
  dataFile.println();
  dataFile.close();
}
}

```

-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN C

### DATASHEET

nRF24L01+ Preliminary Product Specification



#### 2.2 Pin functions

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin. Active low
7	VDD	Power	Power Supply (+1.9V - +3.6V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply Output (+1.8V) for the internal nRF24L01+ Power Amplifier. Must be connected to ANT1 and ANT2 as shown in <a href="#">Figure 29</a> .
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+1.9V - +3.6V DC)
16	IREF	Analog Input	Reference current. Connect a 22kΩ resistor to ground. See <a href="#">Figure 29</a> .
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+1.9V - +3.6V DC)
19	DVDD	Power Output	Internal digital supply output for de-coupling purposes. See <a href="#">Figure 29</a> .
20	VSS	Power	Ground (0V)

Table 1. nRF24L01+ pin function

## 2 Pin Information

### 2.1 Pin assignment

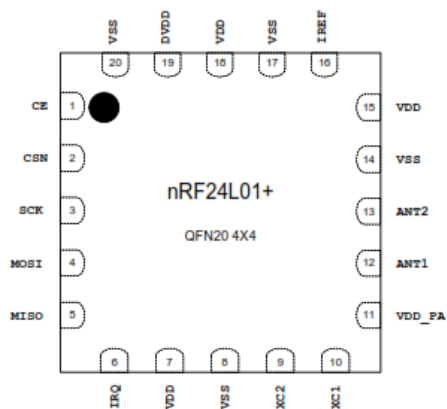


Figure 2. nRF24L01+ pin assignment (top view) for the QFN20 4x4 package



## 1.2 Block diagram

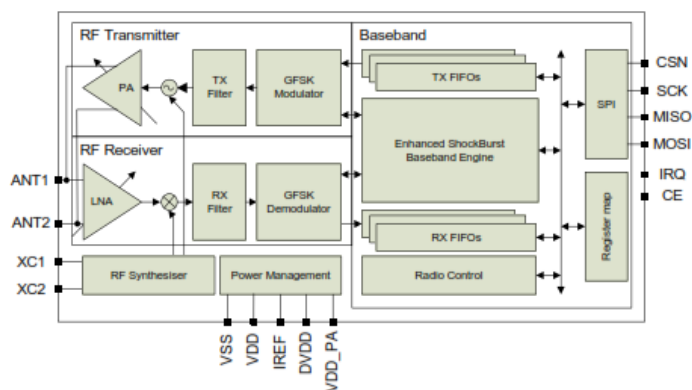
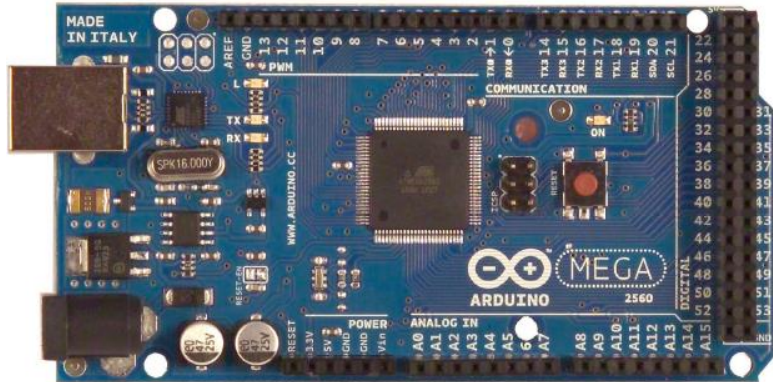
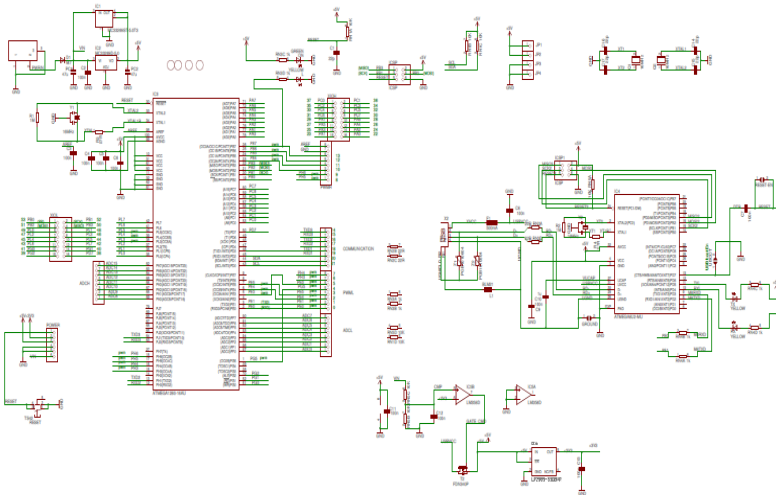


Figure 1. nRF24L01+ block diagram

## Arduino™ Mega 2560 Reference Design

DESIGNED BY ARDUINO S.R.L. FOR THE ARDUINO MEGA 2560. ALL RIGHTS RESERVED. ANY REPRODUCTION OR DISTRIBUTION OF THIS DOCUMENT WITHOUT THE WRITTEN PERMISSION OF ARDUINO S.R.L. IS PROHIBITED. THE INFORMATION CONTAINED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL ARDUINO S.R.L. BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE INFORMATION CONTAINED HEREIN OR THE USE OF THE SOFTWARE.

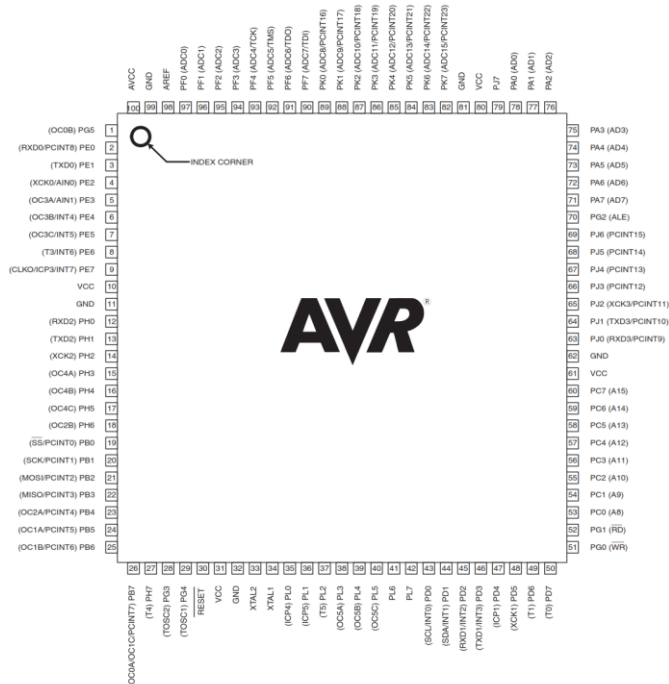


## Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85 °C/ 100 years at 25 °C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
    - Programming Lock for Software Security
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 14 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 816-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-quad OPEN-Drain, 64-lead TQFP (ATmega1281/2561)
  - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
  - RoHS/Full Green
- Temperature Ranges
  - -40 °C to 85 °C Industrial
- Ultra-Low Power Consumption
  - Active Mode: 1MHz, 1.8V, 500uA
  - Power-down Mode: 0.1uA at 1.8V
- Speed Grade:
  - ATmega640V/ATmega1280V/ATmega1281V:
    - 0 - 8MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega2560V/ATmega2561V:
    - 0 - 20MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega640/ATmega1280/ATmega1281:
    - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
  - ATmega2560/ATmega2561:
    - 0 - 16MHz @ 4.5V - 5.5V

## 1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560



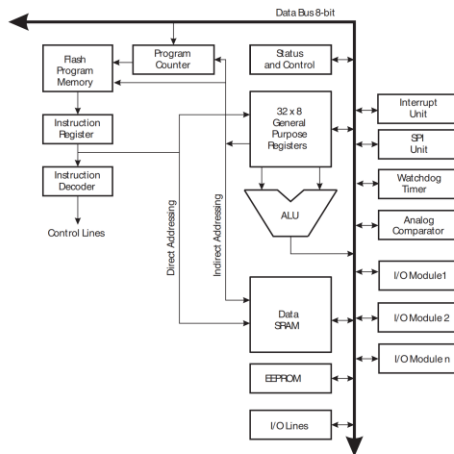
## 7. AVR CPU Core

### 7.1 Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

### 7.2 Architectural Overview

Figure 7-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two oper-





8-bit AVR Microcontrollers

ATmega328/P

DATASHEET COMPLETE

## Introduction

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

## Feature

High Performance, Low Power Atmel®/AVR® 8-Bit Microcontroller Family

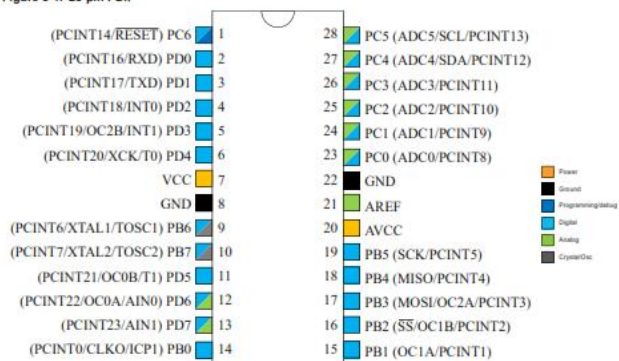
- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix® Acquisition
  - Up to 64 sense channels

Atmel-4272B-ATmega328P\_Datasheet\_Complete-112616

## 5. Pin Configurations

### 5.1. Pin-out

Figure 5-1. 28-pin PDIP



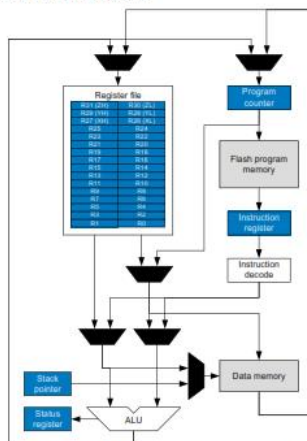


© 2013 Pearson Education, Inc. or its affiliate(s). All rights reserved. This material is intended solely for the personal use of the individual user and is not to be disseminated broadly.

## 11.1. Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 11-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

# HC-05

## -Bluetooth to Serial Port Module

### Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

### Specifications

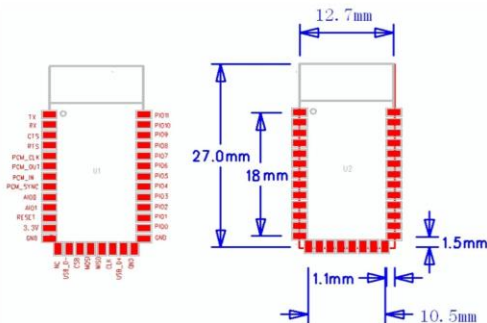
#### Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

## Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

## Hardware



-----Halaman ini sengaja dikosongkan-----

## RIWAYAT HIDUP PENULIS



Nama : Raka Satria Pradana  
H.D  
TTL : Magetan, 4 September  
1996  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat Rumah : Ds. Bulu RT. 3/2 Kec.  
Sukomoro  
Kab.magetan  
Telp/HP : 085804385394  
E-mail : rachasatria92@yahoo.com  
Hobi : Badminton, Desain

### RIWAYAT PENDIDIKAN

- 2002 – 2008 : SDN Bulu 1, Sukomoro
- 2008 – 2011 : SMPN 4 Magetan
- 2011 – 2014 : SMAN 1 Sukomoro
- 2011 – Sekarang : Program Studi Komputer Kontrol,  
Departemen Teknik Elektro Otomasi, ITS

### PENGALAMAN KERJA

- Kerja Praktek PT.PJB UP-Gresik. (Agustus-September 2014)

### PENGALAMAN ORGANISASI

- Staf Departemen Kewirausahaan ITS 2013 – 2014
- Staf *Organizing Committee* Dekorasi IARC 2014
- Staf *Organizing Committee* Dekorasi IARC 2015